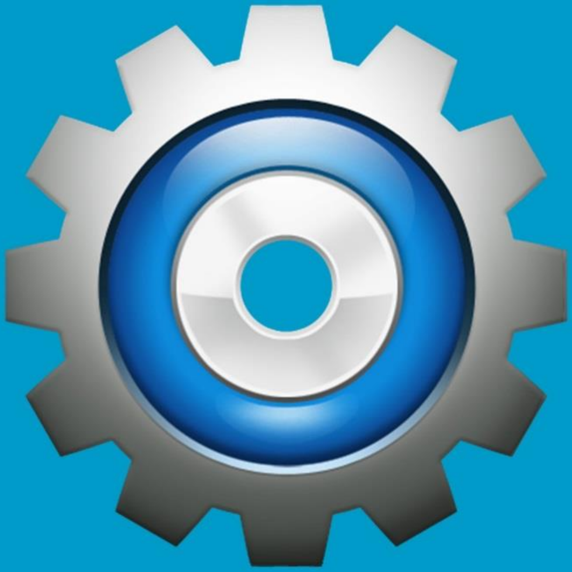
UML ERP Workshop

Writing a BRD with Enterprise Architect

Inventory Control



Jose Zouain

UML - ERP Workshop

Writing a Business Requirement Document (BRD) for the Inventory Control module

Designing an Enterprise Resource Planning (ERP) System with the UML modeling tool, Enterprise Architect by writing a Business Requirement Document for the Inventory Control module.

Jose Zouain

This book is for sale at https://leanpub.com/uml-erpworkshop

This version was published on 2020-JUL-15

Table of Contents

Introduction 6
Copyright and Disclaimer 8
About the Book... 9
About the Author 11
How this Book is Organized 12
1 The Basic Knowledge 13
Enterprise Resource Planning (ERP) 13
Unified Modeling Language (UML)... 14
Business Requirement Document (BRD) 15
Software Design Document (SDD) 16
2 Managing Business Rules with EA 17
Business Rules and OOD 17
Business Rules Benefits.. 17
Business Rules Classification 17
Business Rules in Use Case... 17
Business Rules in Activity Diagram 17
3 Managing Requirements with EA 18
MoSCoW method.. 18
Creating a Requirement Diagram 18
Adding Elements to a Requirement Diagram 18
Connecting Requirement Elements 18
Linking Requirement to a Use Case.. 18
Deleting a Requirement Linked/Not Linked 18
Requirement Relationship Matrix 18
4 Versioning Requirements with EA 19
Requirement for version 5.0 19
Requirement for version 7.4 19

Requirement for version 7.5 19
Requirement for version 7.6 19
5 Enterprise Architect Hints 20
General EA Hints 20
Diagram Hints 20
User Interface Hints 20
Use Case Hints 20
Activity Diagram Hints 20
Business Rule Hints 20
Naming Convention Hints 20
Color Scheme Hints 20
Creating Use Case Scenario and Activity Diagram 20
Breaking the Rule of the Exception Path 20
Breaking the Rule Event and Response 20
6 Organizing and Setting up EA 21
Organizing EA for One Functional Module 21
Organizing EA for more than One Functional Module 21
7 Preface to the BRD 22
Organizing the Use Cases for the Inventory Control 22
Invokes in Use Cases 22
Include in Use Cases 22
Extend in Use Cases 22
Use Case Color Scheme for the Inventory Control 22
Inventory Data for the Inventory Control 22
8 The Inventory Control BRD 23
Cover Page 23
Copyright and Contact Information 23
Introduction 23
Application Overview 23
The Business Requirement Document 23
The Requirements 23
The Use Cases 23
Business Rules 23

Sign-off Page 23
Use Case Diagram sample – Inventory File 24
Use Case Diagram sample – Zeta Pro System 25
Requirement Diagram sample – Inventory Control... 26
Activity Diagram sample – Inventory Line 27
Wireframe sample – IM08.01 Inventory Line.. 28
Zeta System Common Process sample 29

Introduction

Anyone trying to develop a system within their organization has been faced with many competing demands. Demands come from those responsible for establishing the guidance for the organization, the industry standards that are designed to assist in establishing direction, the features provided by the available tool sets in this space, and those who are impacted by the direction as they attempt to do their daily work.

Many standards are large and difficult to implement, but once you have adopted the way you will be working, most standards should make your work easier to do, not harder. I will teach you how I have implemented in different organizations a BRD (Business Requirement Document), making changes as your experience grows using a robust tool, Enterprise Architect from Sparx Systems and Zeta Pro ™ examples. Zeta Pro ™ is an Enterprise Resource Planning System developed by Zeta Concepts, Inc.

Once the main Use Cases and other diagrams are essentially complete, a lot of work remains to be done on the structure of the BRD. There is no intention to release the code of Zeta Pro ™, each developer writes code differently. Coding is like writing; every developer has their own way of coding just as a writer does. Showing the diagrams of Zeta Pro from my point of view is better than showing the code. The system is not perfect and like any software development project, bugs can be found at runtime, unexpected cases or maybe design flaws can occur, but thanks to UML the process of revealing all of this is much easier.

Now that I have talked about the scope of the project I have to mention the tools that I have used: “WithClass” from MicroGold Software; Visual Paradigm from Visual Paradigm International; Rational Rose from IBM Corporation and the tool I have been using for the last sixteen (16) years: Enterprise Architect from Sparx Systems. I agree with the Enterprise Architect manual which states, Enterprise Architect is an intuitive, flexible and powerful UML analysis and design tool for building robust and maintainable software. From requirements gathering, through analysis, modeling, and implementation and testing to deployment and maintenance, Enterprise Architect is a fast, feature-rich, multi-user UML modeling tool, driving the long-term success of your software project.

As you get more comfortable, you will naturally start to find the tool better to work with. There is no shortage of articles about the right or best way to create a BRD, and there are lots of potential approaches; you can learn the concepts by completing inter-actives exercises, UML-ERP Workshop offers you that.

After several years updating I have decided to divide the BRD in system wide functional sections, each section will cover a different part: the **Processes** covers the transaction processes, the **Maintenance** covers the data entry, the **Report** covers how to run reports and finally, the **Settings** covers how to set up parameters for a system. If you still want all of them together you can still get the Full BRD.

Each section will include all the **Business Rules** and the **Zeta System Common Process** which can be used everywhere within the system and it includes the requirements, use cases, activity diagrams and wireframes for that particular section.

This sample will cover the following eBooks:

- Inventory Control – Full BRD (All Sections)
- Inventory Control – Processes Section BRD
- Inventory Control – Maintenance Section BRD
- Inventory Control – Report Section BRD
- Inventory Control – Settings Section BRD

Copyright and Disclaimer

Following you will find the Statement of Confidentiality and Copyright that is included in each BRD document I have created; this applies for this eBook also, included is a Trademark section. I would be glad to receive any kind of feedback to correct future updates of this eBook; all the information presented here is subject to change without notice.

Trademarks

Zeta Pro ™ is a trademark of Zeta Concepts, Inc.

Sparx and the names of Sparx's products referenced in this eBook are trademarks of or registered trademarks of Sparx Systems Pty Ltd. All other products names and logos in this eBook are used for identification purposes only and may be trademark or registered trademarks of their respective companies.

Statement of Confidentiality

The information contained in the following document is proprietary and strictly confidential. It is intended to be reviewed only by the party receiving it from the Zeta Concepts, Inc. ("Zeta") and should not be made available to any person or entity without the written consent of Zeta. Any information contained Zeta makes no warranty or representation, expressed or implied, whatsoever regarding the accuracy or completeness of the information provided.

Copyright

The information in this document may not be reproduced in any written, electronic, recording, or photocopying without express written permission of Zeta. Although every precaution has been taken to verify the accuracy of the information contained herein, Zeta assumes no responsibility for any errors or omissions. No liability is assumed for damages that may result from the use of information contained within.

About the Book

A lot of books have been written about the Unified Modeling Language (UML) with very good examples, although I have never seen a real life example of a working system on the market, but maybe I am wrong. In this eBook I will reveal all the business processes and the business rules on how to create the Inventory Control module of the system; Use Cases and Activity Diagrams needed to create the application, as well as what requirements were originally captured to develop the application.

As a UML evangelist I have decided to show everybody the system I designed and helped develop, Zeta Pro ™. These are open venues for both software developers who are learning how to develop or have been creating systems for quite some time.

Zeta Pro ™ has been in the market for more than 15 years. It is an Enterprise Resource Planning System originally developed in the 90s for the DOS platform and later transferred and developed in an Object Oriented Programming language for the Windows platform, using UML diagrams.

This eBook is for the Business Analysts who will learn how to create a Business Requirement Document (BRD).

Zeta Pro ™ (the system) is huge, big, Grande; therefore I will be releasing several eBooks covering a complete module or section of a module. Also for those readers who want to see the system in action and follow the diagrams, send me an email for a single Zeta Pro ™ user license for 180 days (6 months) free!

Go to my website www.zetaconcepts.com and register to receive a single user license or send me an email to support@zetaconcepts.com requesting the single user license.

The following eBooks will be published:

- Inventory Control – Full BRD (All Sections)
- Inventory Control – Processes Section BRD
- Inventory Control – Maintenance Section BRD
- Inventory Control – Report Section BRD
- Inventory Control – Settings Section BRD
- Zeta Pro ™ – Report Engine SDD

In this eBook and subsequent ones I will show you how to design a system. I did it - why can't you do the same? The only advantage that you will have over me is that no one supplied me with all the UML diagrams to design it as we are doing with all these eBooks.

You will leverage the techniques from these eBooks by discovering how UML can help you identify opportunities for your personal goals.

This is a complete, practical, hands-on book which is comprised of step-by-step instructions for creating an Enterprise Resource Planning System, covering everything you need to know about business processes, business rules, system processing and visualizing data at the Enterprise Level.

About the Author

José Zouain is the original author and principal maintainer of Zeta Pro ™, and has been writing code for about 20 years. As an UML evangelist since 1999 and an Enterprise Architect ™ Guru since 2002, José has been planning and discussing this project with friends for several years, and finally has decided to go ahead and move forward with it. José's commitment to this project and his free time is limited due to working a day job, the available time he has to work on these eBooks, has been evenings and spare time on the weekends.

An independent consultant with over twenty (20) years of experience in different phases of the Software Development Life Cycle (SDLC), managing advance programming, system maintenance and problem determination, he has fulfilled a number of roles. Among his responsibilities he has held different titles such as Senior Technical Analyst and Senior Business Analysts, Project Manager, User Acceptance Manager, and OOP (Object Oriented Programming) Developer.

Online

Website of the author

www.zetaconcepts.com

www.zetapro.com

Email of the author

support@zetaconcepts.com

How this Book is Organized

Chapter 1: “The Basic Knowledge”, presents the concept of Enterprise Resource Planning (ERP) and Unified Modeling Language (UML); also presents what is a Business Requirement Document (BRD). If you are proficient in UML, feel free to skip this chapter. There are, however, some important definitions and sections that highlight some areas that are relevant to the rest of the eBook.

Chapter 2: “Managing Business Rules with EA”, is an overview of what a business rule is and the types of business rules. The structured approach used to record the business rules using “natural” language.

Chapter 3: “Managing Requirements with EA”, is an overview of what a requirement is and how to manage the requirements within EA. Writing good business requirements is like writing good code.

Chapter 4: “Versioning Requirements with EA”, explains a real life example how has the “Assign Serial to Inventory” evolved during the years with different versions within the software, and how EA has track the versioning. For new users to EA skip this chapter for later after you have manage EA for some time.

Chapter 5: “Enterprise Architect Hints”, defines all the hints needed to succeed and all the knowledge accumulated over the last 15 years that will make your life easier using Enterprise Architect as the tool.

Chapter 6: “Organizing and Setting up EA”, introduces how to organize and set up Enterprise Architect for any software development project. Demonstrates the techniques and notation I have used throughout the eBook.

Chapter 7: “Preface to the BRD”, explaining how is the BRD divided and why we have used this approach in our diagram based on hints provided.

Chapter 8: “Inventory Control - BRD”, lastly what everybody wants to see, is the Business Requirement Document for the Inventory Control module that Enterprise Architect generates from all the information entered and diagrams created based on the templates we have designed.

Chapter 9: “Stakeholder Approval”, this is where the person who has requested the change or changes would provide the signoff.

1 The Basic Knowledge

In this chapter we will be briefly touching on the definition of ERP, UML and BRD.

Enterprise Resource Planning (ERP)

ERP is an industry term for the broad set of activities that helps a business manage the important part of its business. ERP software applications can be used to manage product planning, parts purchasing, inventories, interacting with suppliers, providing customer service, and tracking orders, it can also include application modules for financial accounting and fixed assets management. Typically the ERP system is integrated with a relational database system.

Enterprise Resource Planning Systems are typically developed based on a large framework. In reality, this framework is created and designed by different users (analyst, architects, developers, data modelers) in providing a solution to a customer. It is therefore of interest to make the development process as efficient as possible and one way to do this is to provide them with the specialized models. This book presents our ideas for such domain specific modelling environments, where the graphical notation is inspired by UML, but incorporates specific aspects unique to the specific framework we are considering.

IT leaders are fed up with the perpetual-license-plus-maintenance approach to enterprise software purchases, and they are tired of running on the “upgrade” treadmill simply for the sake of keeping their software maintenance status current. Here is a look at how you can break free and develop your own ERP system, with all the screens, business rules, data models and processes; and use it either internally or in the cloud.

The deployment of an ERP system can involve considerable business process analysis, employee retraining and new work procedures; you can’t simply turn it on and expect it to run without training.

You can use the diagrams to visualize the system from different perspectives, as no complex system can be understood in it’s entirely from one perspective. Diagrams are used for communication, for example, a class may appear on one or more Class Diagrams; it might be represented in a State Machine Diagram, and have instances appeared on a Sequence Diagram, and each diagram will provide a different perspective.

Unified Modeling Language (UML)

The Unified Modeling Language (UML) is a general-purpose modeling language in the field of software engineering which is designed to provide a standard way to visualize the design of a system.

It was created and developed by Grady Booch, Ivar Jacobson and James Rumbaugh at Rational Software during 1994–95, with further development led by them through 1996. In 1997 it was adopted as a standard by the Object Management Group (OMG), and has been managed by this organization ever since. In 2005 the Unified Modeling Language was also published by the International Organization for Standardization (ISO) as an approved ISO standard. Since then it has been periodically revised to cover the latest revision of UML.

A lot of books have been written about UML; however we will briefly touch on the following topics as needed for this project, defining them in simple way:

- **Functional Requirement** – What a system must do, the functions it should perform.
- **Non-Functional Requirement** – What the system must be, it describes the system attributes.
- **Use Case** - Is a sequence of actions or steps performed by an actor and the system, which yields an observable result. A Use Case describes the functionality that will be built in a proposed system, for example adding an Inventory, updating an Inventory and deleting an Inventory are all Use Cases combined into one Use Case.
- **Actors** – Anything that interfaces with the system, a human or machine entity that interacts with the system to perform meaningful work.
- **Activity Diagram** – Is a graphical representation of the flow of steps involved in the execution of a system.
- **Pre-Condition** – Is the state that the system must be in for the use case to be able to start.
- **Post-Condition** - Lists possible states that the system can be in at the end of the use case execution.
- **Trigger -** Is the initiator of a use Case, it is what causes the Use Case to start.
- **Basic Path** – Also called the Happy Path is an unconditional set of steps that describe how the use case goal can be achieved and all related stakeholder interests can be satisfied.
- **Alternate Path** - Is a step or a sequence of steps that achieves the use case’s goal following different steps than described in the main success scenario. But the goal is achieved finally. RESULT POSITIVE.
- **Exception Path** - Is anything that leads to NOT achieving the use case’s goal. RESULT NEGATIVE. Just to be clear: Cancel is no Alternate Path, it’s an Exception Path.
- **Includes relationship** - A relationship between two use cases in which one use case 'includes' the behavior. This is indicated where there a specific business use cases which are used from many other places.
- **Extends relationship** - A relationship between two use cases in which one use case 'extends' the behavior of another. Typically this represents optional behavior in a use case scenario, which is adding behavior to a use case without changing the original use case.
- **Invokes relationship -** Indicates that a use case at some point causes another use case to happen. One use case invokes another in the same way a function invokes a peer function.

- **Precedes relationship** - Indicates that a use case must be completed before another use case can begin. Indicates that one use case needs to precede another within a logical sequence.

Note: To see these four types of use case relationship in a single use case diagram, see **UIM02 – Inventory Use Cases**, in the **Full BRD** or in the **Maintenance Section BRD**.

Business Requirement Document (BRD)

A Business Requirement Document is also called Requirement Definition Document (RDD) is a list of the business requirements with emphasis on “what” is required to deliver a software or a system, rather than on how to achieve it.

The process of creating a BRD consists of the following steps:

1. Gathering and eliciting the system requirements
2. Entering the requirements in the UML tool, Enterprise Architect
3. Define the user interface or wireframe base on requirements
4. Create the use cases of the different components of the system
5. Create the activity diagram per use case
6. Define the business rules and include in use case and activity diagram

Steps 4 and 5 are can be reversed, meaning that the activity diagram can be created first in order to understand the flow of the system; also at step 6 the business rules are captured when gathering requirements but are structured defined later on.

A BRD contains the following sections:

- Cover page
- Copyright and Contact Information
- Introduction
- Glossary (optional)
- Application Overview
- Revision History
- System Impact
- Requirements
- User Interface
- More to follow in the eBook…

Software Design Document (SDD)

A Software Design Document (SDD) also called Software Design Description or Software Design Specification, is a written description of a software product that a software designer or architect writes in order to give a software development team overall guidance to the architecture of the software project. There are two kinds of design documents called High Level Design Document (HLDD) and the Low Level Design Document (LLDD). When writing a software design document, your goal is not to impress the stakeholders; it is to clearly and accurately describe your solution to your team. The best way to achieve that is to keep your language plain and illustrative. Use simple words, short sentences, bullet lists, and helpful diagrams wherever you can. Get everyone involved in your team and treat the SDD as a living document, by updating the documentation as you make changes to the original solution and keeping all the stakeholders in the same page.

An SDD contains the following sections:

- Cover page
- Copyright and Contact Information (optional)
- Introduction (Purpose, Glossary, Overview, Scope, References, Timelines, Revision)
- Architectural Design
- Interface Design
- More to follow in the eBook…

2 Managing Business Rules with EA

Business Rules are statements that define or constrain some aspect of the business. They assert business structure to control or influence the behavior of the business. Implementing business rules can be a tedious and error-prone task in application development.

- Business rules are often ambiguously defined and recorded.
- Business rules are usually a major task in each project, introducing a significant risk to the success of your application.

This lead to a number of problems:

- Difficult to estimate the time required for the implementation
- Developers may misinterpret the requirements
- Complex rules involving multiple objects are often partially implemented
- Difficult to determine if all rules have been implemented.

Topics for this chapter:

Business Rules and OOD

Business Rules Benefits

Business Rules Classification

Business Rules in Use Case

Business Rules in Activity Diagram

3 Managing Requirements with EA

Project begins by gathering requirements from various stakeholders (people who have relevant interest in the old or new systems). A key task for the business analyst is to collect and refine the requirements and agree with the business that they represent exactly what they want of the new or modified system.

Writing good requirements is very simple, it requires three steps:

1. Put yourself in the user’s shoes and understand how they will use the product.
2. Address the 5Ws – Who, What, Where, When, and Why.
3. Break down the requirements into simple parts – do not complicate it.

Topics for this chapter:

MoSCoW method

Creating a Requirement Diagram

Adding Elements to a Requirement Diagram

Connecting Requirement Elements

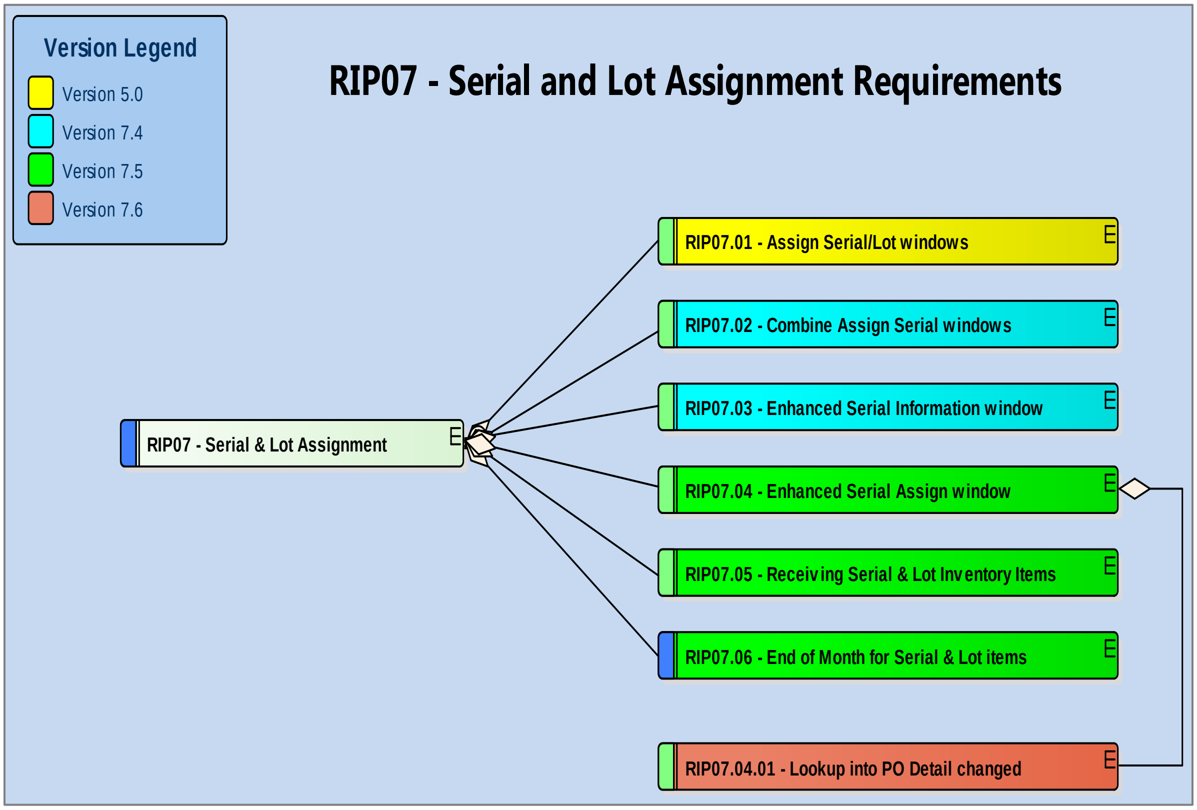
Linking Requirement to a Use Case

Deleting a Requirement Linked/Not Linked

Requirement Relationship Matrix

4 Versioning Requirements with EA

Following will show how to manage requirements inside EA by describing the different steps to follow and a real life example when we had to enhance the “Assign Serial to Inventory” option adding new requirements to the existing requirements and how to proceed with this task. Following is the requirement diagram **RIP07 – Serial and Lot Assignment Requirements,** displaying the requirements from four different versions. For version 5.0 is when the “Assign Serial to Inventory” option was introduced, then three (3) new releases were introduced for version 7.4, 7.5 and 7.6. We will show in this chapter up to version 7.5; for version 7.6, the last requirement, is included at the Inventory Control **Full BRD** or in the **Process Section BRD** as **IP07 – Assign Serial to Inventory**. Unless asked by the stakeholder to include all requirements from day one, only include the latest requirement gathered for the BRD.


Version Legend
🟨 Version 5.0
🟦 Version 7.4
🟩 Version 7.5
🟥 Version 7.6
RIP07 - Serial and Lot Assignment Requirements
RIP07 - Serial & Lot Assignment
RIP07.01 - Assign Serial/Lot windows
RIP07.02 - Combine Assign Serial windows
RIP07.03 - Enhanced Serial Information window
RIP07.04 - Enhanced Serial Assign window
RIP07.05 - Receiving Serial & Lot Inventory Items
RIP07.06 - End of Month for Serial & Lot items
RIP07.04.01 - Lookup into PO Detail changed

Topics for this chapter:

Requirement for version 5.0

Requirement for version 7.4

Requirement for version 7.5

Requirement for version 7.6

5 Enterprise Architect Hints

Following are the EA hints that I can give to you when creating a BRD. The goal of this eBook is to give you the essentials so that you will know how to read UML, use UML and create UML based diagrams. More information can be found in other books available on UML; but you can grasp the visual world of UML and how you can apply this knowledge to your projects based on what you see here.

Topics for this chapter:

General EA Hints

Diagram Hints

User Interface Hints

Use Case Hints

Activity Diagram Hints

Business Rule Hints

Naming Convention Hints

Color Scheme Hints

Creating Use Case Scenario and Activity Diagram

Breaking the Rule of the Exception Path

Breaking the Rule Event and Response

6 Organizing and Setting up EA

The goal of using Enterprise Architect is to provide a central repository that details the functionality of the software developed by the company. Business analyst as well developers, architects, data modelers will utilize EA to document the software from the requirements and the use cases to the databases and the classes required to implement the software.

You can organize the Project Browser in different ways; there are two approaches that I like:

- **Diagram section** – organized by diagrams, like Requirements, Use Cases, Business Rules, Class Model, Data Model and so on.
- **Functional section** – organized by functional section or functional module, is the hierarchy that we will be using in this eBook.

Topics for this chapter:

Organizing EA for One Functional Module

Organizing EA for more than One Functional Module

7 Preface to the BRD

In the following chapter will be showing the Business Requirement Document (BRD) for the Inventory Control module for the Zeta Pro ™ system, developed by Zeta Concepts, an Enterprise Resource Planning System with eight (8) modules, in this BRD we are implementing one module only.

Zeta Pro ™ system modules:

1. General Ledger
2. Sales and Receivables
3. Purchase and Payables
4. **Inventory Control**
5. Bank Reconciliation
6. Contact Relation
7. Asset Management
8. Query Management

Organizing the Use Cases for the Inventory Control

Following the organization schema from the chapter ***Organizing and Setting up EA***, stating that every module must have one use case "entry point"; and every sub-system must also have one use case "entry point"; the Inventory Control module as all the other modules has four (4) sub-systems entry point:

- Process Administration
- Maintenance Administration
- Reports Administration
- Settings Administration

Invokes in Use Cases

Include in Use Cases

Extend in Use Cases

Use Case Color Scheme for the Inventory Control

Making use of the Color Scheme Hints from the chapter ***Enterprise Architects Hints***

Inventory Data for the Inventory Control

8 The Inventory Control BRD

Topics for this chapter:

Will show the user how to write a Business Requirement Diagram (BRD), different sections of a BRD and what to include in each section with real-life example of a system with 20 years in the market:

Cover Page

Copyright and Contact Information

Introduction

- Purpose of the BRD
- Glossary

Application Overview

- The Scope
- The Context

The Business Requirement Document

- Revision History
- Table of Figures

The Requirements

- Requirement Diagram
- Requirement Specification

The Use Cases

- Use Case Diagram
- Use Case Scenario
- Activity Diagram
- Wireframe
- Zeta System Common Process

Business Rules

- Operatives Rules
- Metric Rules
- Log Event Rules

Sign-off Page

Use Case Diagram sample – Inventory File

Note: The four types of a use case relationship in a single use case diagram:

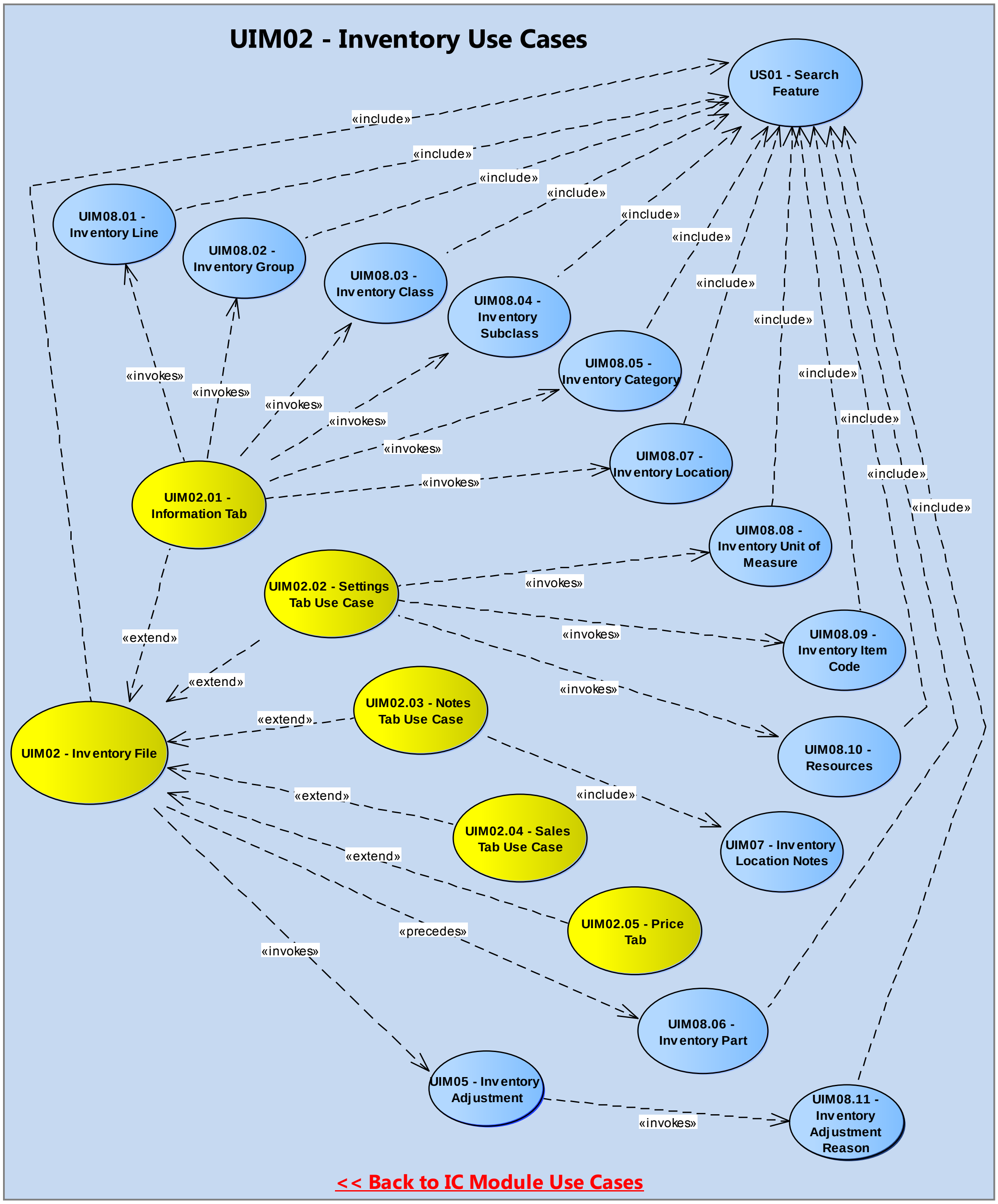
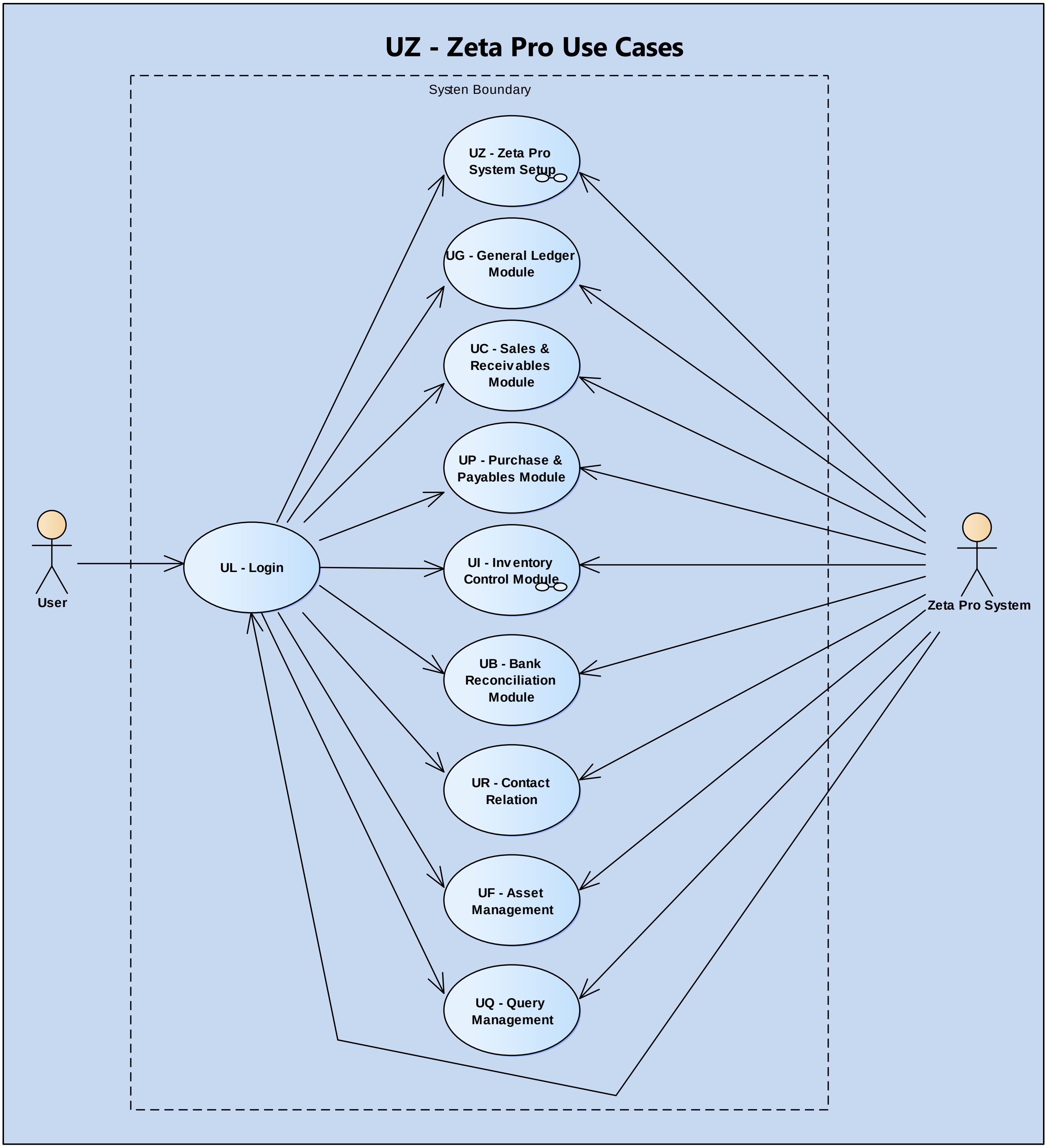
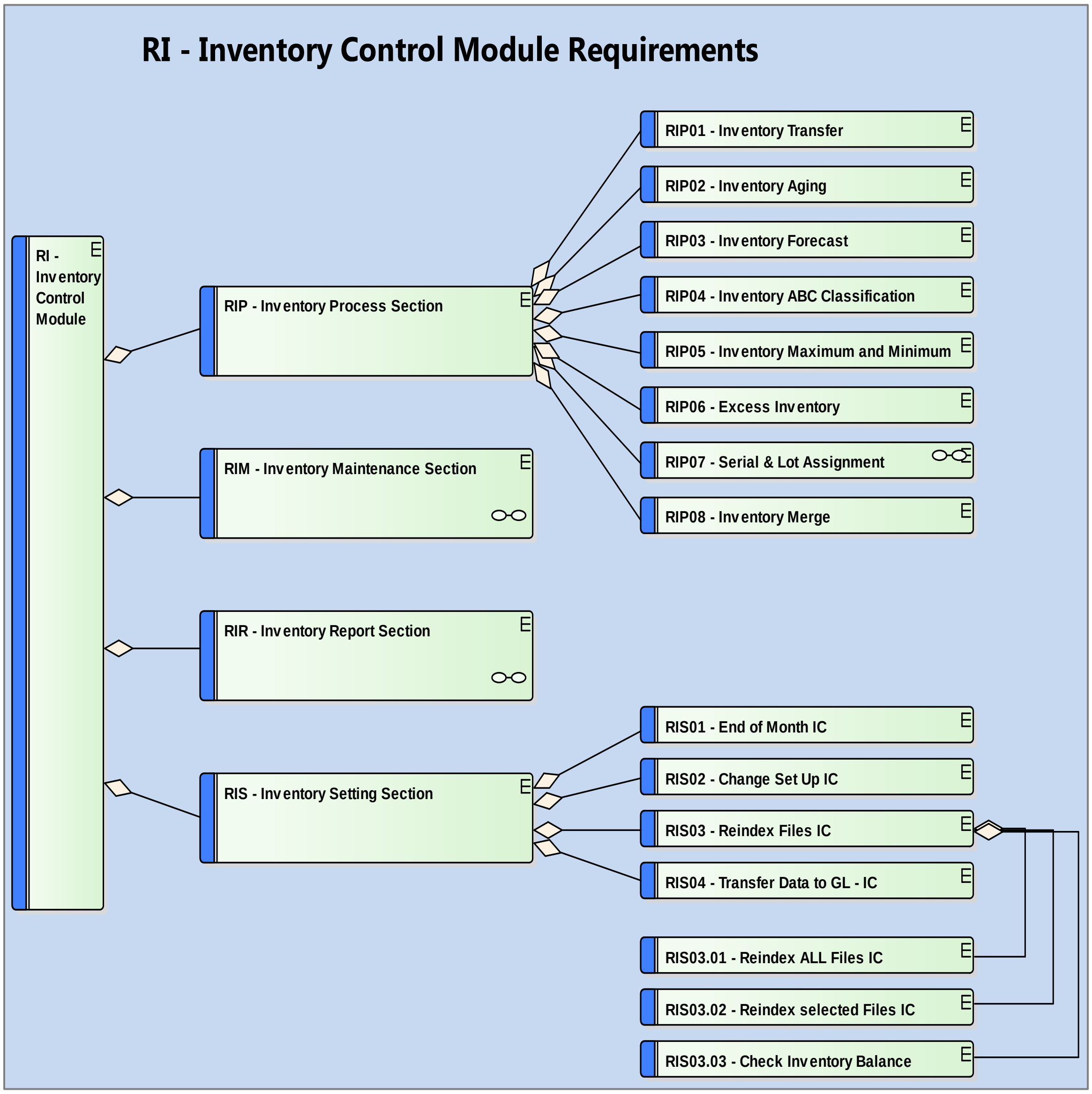

UIM02 - Inventory Use Cases
US01 - Search Feature
UIM08.01 - Inventory Line
UIM08.02 - Inventory Group
UIM08.03 - Inventory Class
UIM08.04 - Inventory Subclass
UIM08.05 - Inventory Category
UIM08.07 - Inventory Location
UIM08.08 - Inventory Unit of Measure
UIM08.09 - Inventory Item Code
UIM08.10 - Resources
UIM07 - Inventory Location Notes
UIM08.06 - Inventory Part
UIM08.11 - Inventory Adjustment Reason
UIM05 - Inventory Adjustment
UIM02.01 - Information Tab
UIM02.02 - Settings Tab Use Case
UIM02.03 - Notes Tab Use Case
UIM02.04 - Sales Tab Use Case
UIM02.05 - Price Tab
UIM02 - Inventory File
«include»
«invokes»
«extend»
«precedes»
<< Back to IC Module Use Cases

Figure 20: UIM02 - Inventory Use Case

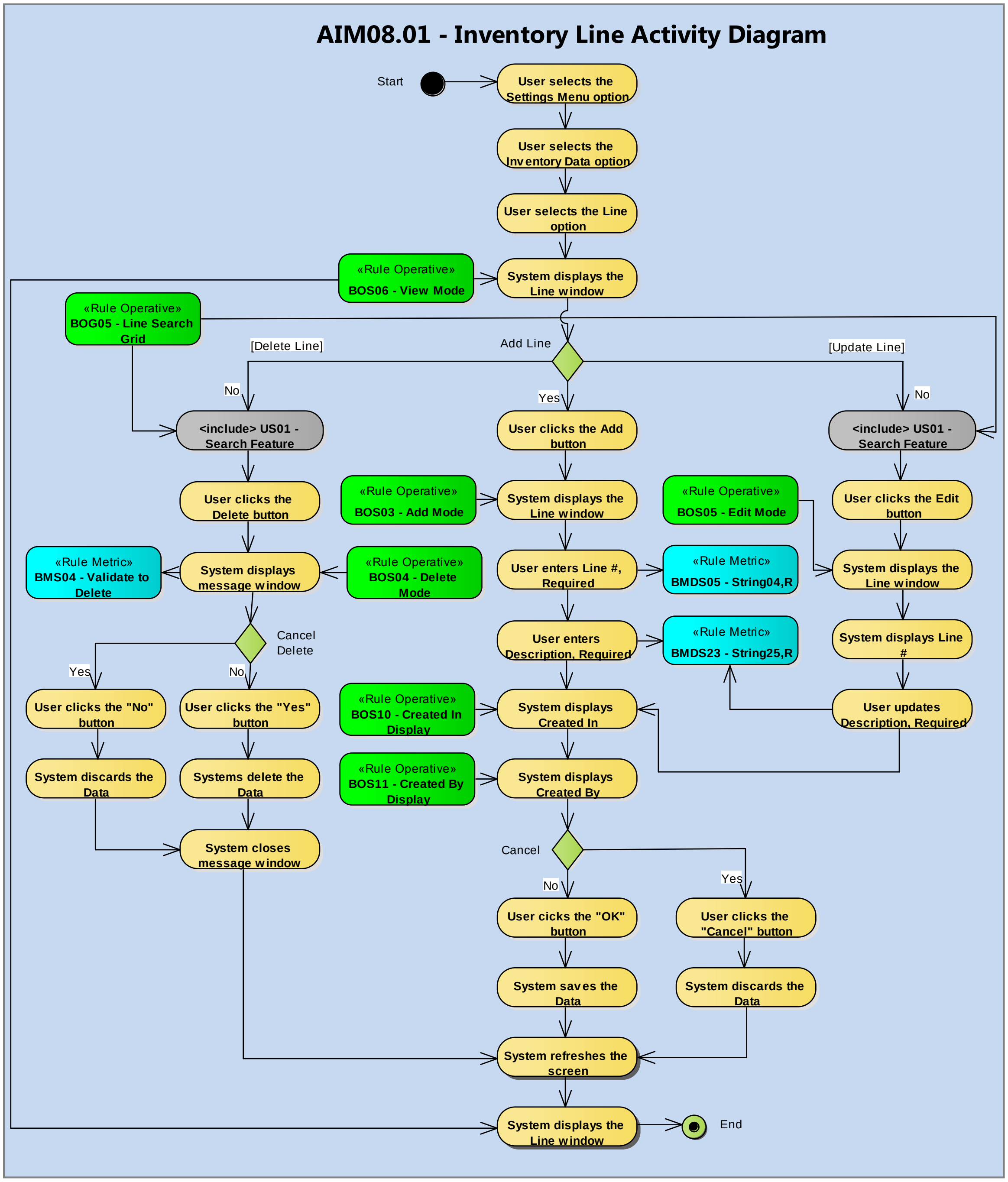
Use Case Diagram sample – Zeta Pro System


UZ - Zeta Pro Use Cases
Systen Boundary
UZ - Zeta Pro System Setup
UG - General Ledger Module
UC - Sales & Receivables Module
UP - Purchase & Payables Module
UI - Inventory Control Module
UB - Bank Reconciliation Module
UR - Contact Relation
UF - Asset Management
UQ - Query Management
UL - Login
User
Zeta Pro System

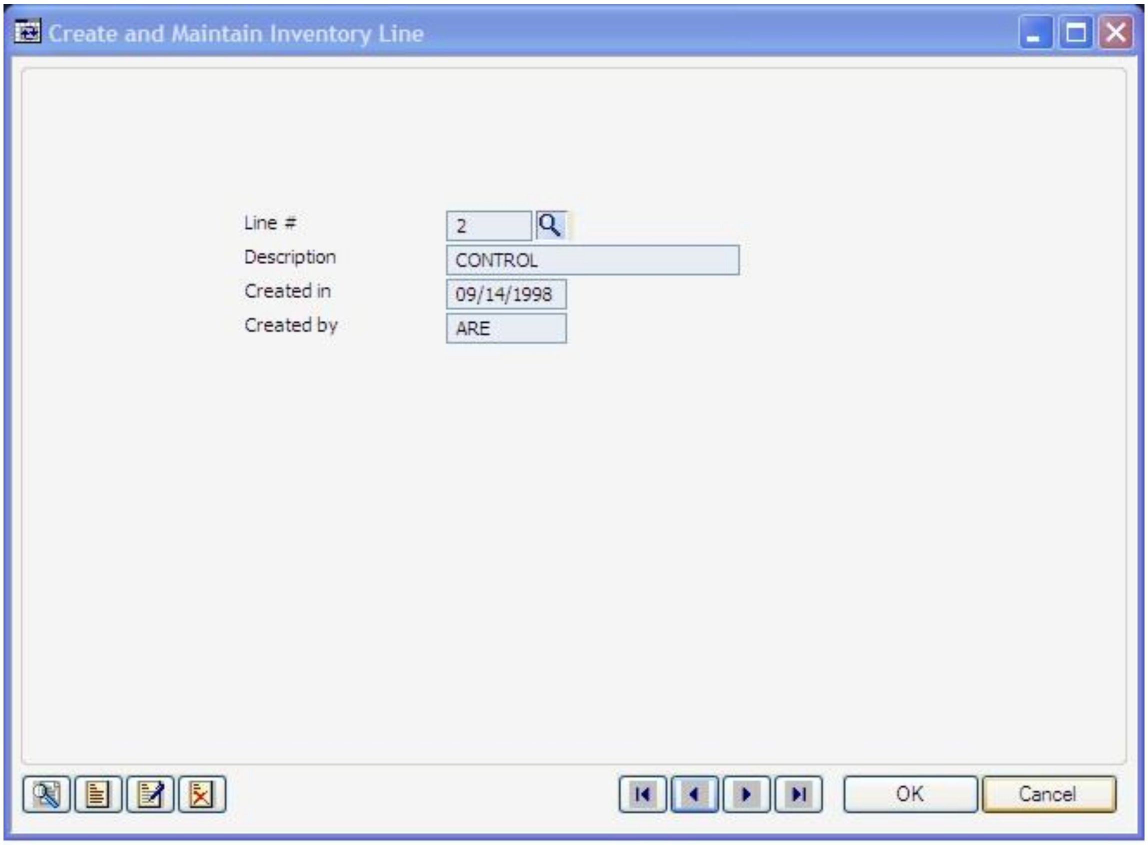
Requirement Diagram sample – Inventory Control


RI - Inventory Control Module Requirements
RI - Inventory Control Module
RIP - Inventory Process Section
RIM - Inventory Maintenance Section
RIR - Inventory Report Section
RIS - Inventory Setting Section
RIP01 - Inventory Transfer
RIP02 - Inventory Aging
RIP03 - Inventory Forecast
RIP04 - Inventory ABC Classification
RIP05 - Inventory Maximum and Minimum
RIP06 - Excess Inventory
RIP07 - Serial & Lot Assignment
RIP08 - Inventory Merge
RIS01 - End of Month IC
RIS02 - Change Set Up IC
RIS03 - Reindex Files IC
RIS04 - Transfer Data to GL - IC
RIS03.01 - Reindex ALL Files IC
RIS03.02 - Reindex selected Files IC
RIS03.03 - Check Inventory Balance

Activity Diagram sample – Inventory Line


AIM08.01 - Inventory Line Activity Diagram
Start
User selects the Settings Menu option
User selects the Inventory Data option
User selects the Line option
«Rule Operative» BOS06 - View Mode
System displays the Line window
«Rule Operative» BOG05 - Line Search Grid
[Delete Line]
Add Line
[Update Line]
No
Yes
No
<include> US01 - Search Feature
User clicks the Add button
<include> US01 - Search Feature
User clicks the Delete button
«Rule Operative» BOS03 - Add Mode
System displays the Line window
«Rule Operative» BOS05 - Edit Mode
User clicks the Edit button
«Rule Metric» BMS04 - Validate to Delete
System displays message window
«Rule Operative» BOS04 - Delete Mode
User enters Line #, Required
«Rule Metric» BMDS05 - String04,R
System displays the Line window
Cancel Delete
User enters Description, Required
«Rule Metric» BMDS23 - String25,R
System displays Line #
Yes
No
User clicks the "No" button
User clicks the "Yes" button
«Rule Operative» BOS10 - Created In Display
System displays Created In
User updates Description, Required
System discards the Data
Systems delete the Data
«Rule Operative» BOS11 - Created By Display
System displays Created By
System closes message window
Cancel
No
Yes
User cicks the "OK" button
User clicks the "Cancel" button
System saves the Data
System discards the Data
System refreshes the screen
System displays the Line window
End

Wireframe sample – IM08.01 Inventory Line



Line # 2

Description CONTROL

Created in 09/14/1998

Created by ARE

WIM08.01 - Inventory Line Screen

WIM08.01 - Inventory Line Screen
To add, update or delete an Inventory Line, the user must select the “Line” option from the Inventory Data of the Settings menu option, the system will start the "Create and Maintain Inventory Line" window, the user will then search for the Line, which will display relevant data.

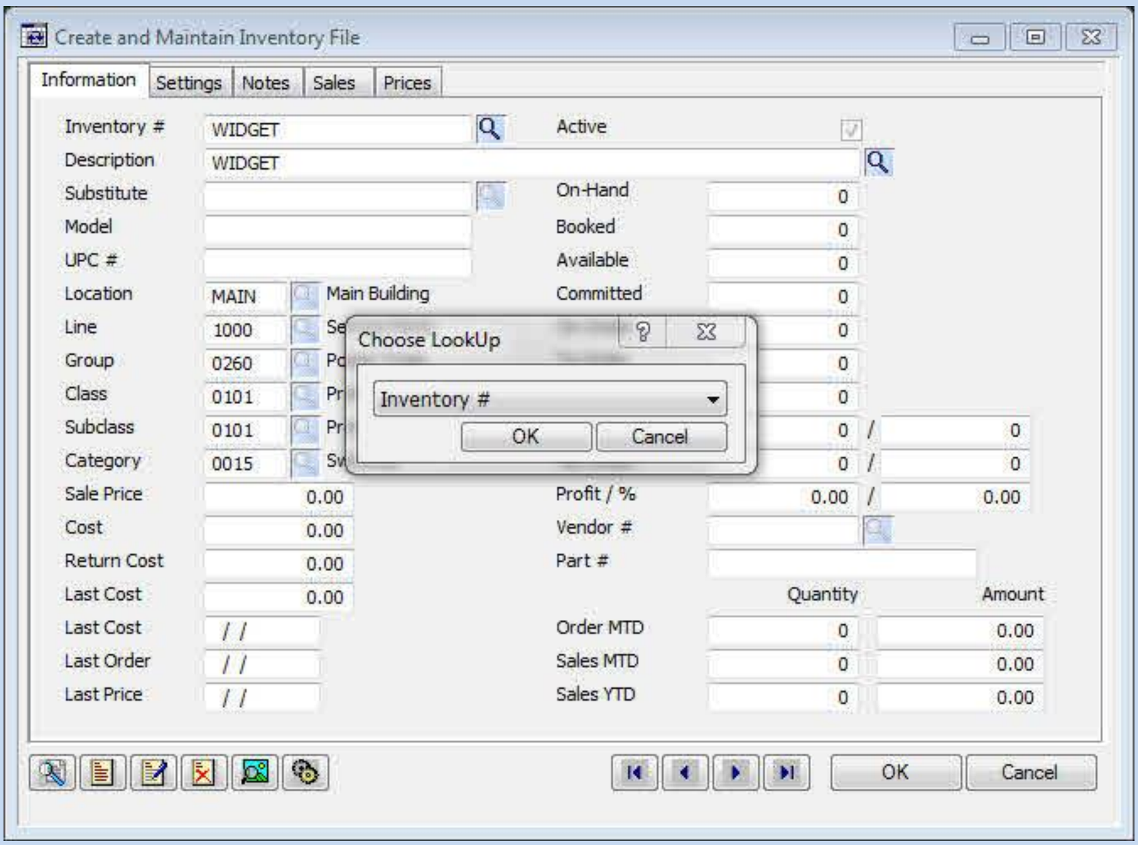
Zeta System Common Process sample

ZS – Zeta System Common Process

This section includes the system common process for the Zeta Pro System, these processes are those used globally or everywhere in the system.

- Search Feature
- Box Lookup Feature
- User Defined Report
- Report Destination
- Box Zoom Lookup Feature

WS01 – Search Feature Screen



Inventory # WIDGET
Active ☑
Description WIDGET
Substitute
Model
UPC #
Location MAIN Main Building
Line 1000
Group 0260
Class 0101
Subclass 0101
Category 0015
Sale Price 0.00
Cost 0.00
Return Cost 0.00
Last Cost 0.00
Last Cost / /
Last Order / /
Last Price / /
On-Hand 0
Booked 0
Available 0
Committed 0
Profit / % 0.00 / 0.00
Vendor #
Part #

Choose LookUp: Inventory #

	Quantity	Amount
Order MTD	0	0.00
Sales MTD	0	0.00
Sales YTD	0	0.00

WIS01 - Search Feature Screen

UML - ERP Workshop

An approach that will change how to write Business Requirement Document (BRD) in the Enterprise.

Anyone trying to develop a system within their organization has been faced with many competing demands. Demands come from those responsible for establishing the guidance for the organization, the industry standards that are designed to assist in establishing direction, the features provided by the available tool sets in this space, and those who are impacted by the direction as they attempt to do their daily work.

Many standards are large and difficult to implement, but once you have adopted the way you will be working, most standards should make your work easier to do, not harder. I will teach you how I have implemented in different organizations a BRD (Business Requirement Document) making changes as your experience grows using a robust tool, Enterprise Architect from Sparx Systems, and Zeta Pro examples.

This is a complete, practical, hands-on book which is comprised of step-by-step instructions for creating an Enterprise Resource Planning System, covering everything you need to know about business processes, business rules, system processing and visualizing data at the Enterprise Level.