

Use Case Diagram in Expanded Format



Español



Entries

- High Level Use Case Diagram
 - ◆ Can it be used configuration management?: **Yes**
- Requirements Specification Document
 - ◆ Can it be used configuration management?: **Yes**



Exit

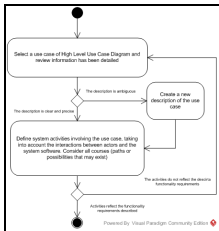
- Use Case Diagram in Expanded Format (using the [Craig Larman Method](#))
 - ◆ Can it be used configuration management?: **Yes**



Solution



Process



Development time

- ◆ To acquire the necessary knowledge to develop the software product: 2 hours
- ◆ To create the Product Pattern: 2 hours
- ◆ To apply the Product Pattern: 10 hours.



Explanatory Video

- ◆ Not applicable



Related Patterns

- [Use Case Diagram](#)
- [High Level Use Case Diagram](#)
- [Requirements Specification](#)





Quality Controllers

- None




Templates

-  [Use Case Information in Expanded Format](#)
-  [Template Use Case Diagram in Expanded Format](#)



Examples

-  [Example use case diagram in expanded format "Login", "LogOut" and "ModifyData" in the context of a social network.](#)



Support Tools

- [Argo UML \(Open Source Project\)](#)
- [Dia \(GNOME\)](#)
- [Rational Software Modeler \(IBM\)](#)
- [StarUML \(Open Source Project\)](#)
- [Visual Paradigm for UML \(Visual Paradigm\)](#)



Initial Context

It is developing a software system using the [Craig Larman Method](#). It is already created the use case diagram high-level system.



Result Context

It has been explained in detail all use cases of the use case diagram of the system, this explanation contains both, information in the format described at a high level, and a description of activities involving each use case including its alternative courses . It has been using the format proposed in [Craig Larman Method](#).



Problem

It is desired to explain in detail all use cases diagram, including a description of system activities involving each.



Restrictions (*Forces*)

- **Type of Organization:** SMEs, Large Companies.
- **System Type** It applies to all types of systems.
- **Programming Paradigm:** OO (Object Oriented).



Roles

- Analyst
- Customer
- Project Manager
- System Users



Lessons Learned

- Benefits of using this pattern
 - ◆ A graphic description of how the software system works and how it will be used is provided.
 - ◆ It is possible to describe the interactions between the system and its actors.
 - ◆ It has a detailed and explicit description of the purpose of each use case diagram.
 - ◆ The activities described are useful for creating the class diagram.



Capability Level

- Capability Level 2 CMMI.



Basic Knowledge and Skills



Knowledge

- Definition of software requirements (functional and non-functional).
- [Craig Larman Method](#)
 - ◆ Notions of UML, especially the technique of modeling of use cases.



Abilities

- Capacity of abstraction.



Information Resources

- Ferré Grau, X & Sanchez-Segura, M. (2004). *Desarrollo Orientado a Objetos con UML*. Recuperado el 2009-11-26 de <http://www.clikear.com/manuales/uml/index.aspx>.
 - Fowler, M. (2004). *UML distilled: a brief guide to the standard object modelling language*. Addison-Wesley.
 - Larman, C. (2001). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process* (2da. ed.). Prentice Hall.
 - Jacobson, I. & Booch, G.(1999). *The unified software development process*. Addison-Wesley.
 - Rumbaugh, J. & Jacobson, I. (2005). *The unified modeling language reference manual*. Addison-Wesley.
 - Rumbaugh, J. (2005). *Object-Oriented Modelling and Design*. Prentice Hall.
-