

## Simplified Requirement Capturing with VP-UML

### Use Case Modeling Made Easy

Visual Paradigm for UML (VP-UML) is a award-winning modeling tool which can help analysts to perform requirement modeling **faster, easier and better**. It is a team-enabled requirement capturing tool that facilitates system analysts and software developers to specify a more comprehensive and accurate set of requirements in **less time** and **match the customers expectation**.

Bear in mind that, the most common reason for software development projects to be slipped in schedule, buggy releases missing features, over budget, or not at all is due to the poorly defined requirements. Don't let this happen again in your team - try Visual Paradigm for UML now!

See the following table for the highlight in a list of selected features and benefits of VP-UML.

Key Features	Benefit
<b>Use Case Identification Textual Analysis</b>	Identify use cases with domain analysis. Visual Paradigm for UML provide an automated textual analysis guided process for streamlining use case identification process.
<b>Use Case Identification Business Process</b>	Understanding complex business workflow with BPM and discover use cases in it. Analysts can better understand the entire business activities (problem space) and identify use cases (solution space) from which the parts of activities planned to be automated.
<b>Authoring Use Case Description</b>	Detailing use case with flow of events in use case description and subsequently sequence diagrams can be generated automatically based on these descriptions.
<b>Diagramming</b>	Visual Paradigm's cutting edge GUI allow software analysts to capturing requirements faster, better and easier, significantly improving the productivity of your entire development. Use Case modeling is an integral part seamlessly embedded within VP-UML.
<b>Publishing and Reporting</b>	Publish your work in popular format such as Word and PDF and/or website for sharing and improving the communication within your team.
<b>Teamwork Support</b>	Multiple versions of development work flow within a team is always a hard problem for us to deal with. With the Teamwork Server, your can now get peace of mind for handing these variant versions of works automatically by the Team Server.

# Why Use Case Modeling

One of the most challenging things to do when developing software applications is to build the right thing, especially for those complex and large-scale systems. Most of the projects failure ending up with expensive rework, delayed releases, or even abandonment are due to the poorly conducted requirement works in the early stage. Requirement specifications should always focus on the users of the system, not the system itself, thus the real system needs are brought to light early on. Use case driven approach encourage designers to envision outcomes before attempting to specify outcomes, and thereby they help to make requirements more proactive in system development.

Since a use case consists mainly of narrative text, it is easily understandable by all stakeholders including customers, end users and managers, not just developers and testers. It builds a shared vision of the targeted systems under developed, by bridging the gap between the people who has the best knowledge of what goals needed to be achieved and the people who understand how to build a solution; not only promoting a sense of ownership from targeted users, but also eliminating surprises when the system is delivered.

Visual Paradigm for UML (VP-UML) provide an intuitive all-in-one development platform which come with the automated Use Case Modeling Toolset as an integral component of the product, enabling use case specification to be further refined into other analysis and design deliverables. Our outstanding automated Use Case Modeling process will help your team to accurately transform systems requirement into quality software solutions, with minimum risk and maximum ROI. We design and develop products that remove complexity, improve productivity, and compress your software development time frames.

***Do wait! Download Visual Paradigm for UML to enjoy the following benefits regardless of which role you are playing in the development team:***

- Easier and earlier user validation.
- Use cases are useful for scoping. Use cases make it easy to take a staged delivery approach to projects; they can be relatively easily added and removed from a software project as priorities change.
- Use cases can also serve as the basis for the estimating, scheduling, and validating effort and they are reusable within a project. The use case can evolve at each iteration from a method of capturing requirements, to development guidelines to programmers, to a test case and finally into user documentation.
- Use case modeling provide basic groundwork for the requirements document, user manual and test cases. Test case scenarios or Test Scenarios can be directly derived from use cases.
- Use cases are concerned with the interactions between the user and the system. They make it possible for user interface designers to become involved in the development process either before or in parallel with software developers.
- Better traceability through the system development process.
- Uniquely identified use cases can be traced back to business requirements or stakeholder needs.
- Use case partitioning can be used to organize and structure the requirements model, • permitting common behavior to be factored out for system reuse.
- Use case alternative paths capture additional behavior that can improve system robustness.
- Good way to start identifying objects from scenarios.
- Helps technical writers in structuring the overall work on the user manuals at an early stage.

# Features

### UML Support

1. Class diagram
2. Use case diagram
3. Sequence diagram
4. Communication diagram
5. State machine diagram
6. Activity diagram
7. Component diagram
8. Deployment diagram
9. Package diagram
10. Object diagram
11. Composite structure diagram
12. Timing diagram
13. Interaction overview diagram
14. Use case detail editor
15. Use case flow-of-events listing
16. Generate sequence diagrams from flow of events lists
17. Select attribute's getter or setter as call message's action
18. Business model use case support

### Requirements Management

1. Requirement diagram
2. Textual analysis
3. CRC Cards
4. User interface designer
5. Identify candidate activity and action by textual analysis
6. Define and customize requirement types
7. Display full set of requirements in tabular view
8. Support generating ID for Requirements

### Business Process Modeling

1. Business process diagram
2. Data flow diagram
3. Event-driven process chain diagram
4. Process map diagram
5. Export business process diagrams to BPEL
6. Identify candidate business process elements using textual analysis
7. Automatically stretch pools and lanes to fit diagram
8. Smart routing for connecting objects
9. Extend business process model with stereotype and tagged value
10. Set state for data object

## Use Case Modeling

### Business Process Modeling

11. Organization chart
12. Relocate a branch of unit through drag and drop
13. Nested lanes support

### Mind Mapping

1. Mind map diagram
2. Create link relationships between nodes
3. Smart layout for mind mapping nodes and diagram

### Database Modeling

1. Entity relationship diagram
2. ORM diagrams
3. Reverse engineer existing databases to entity relationship diagrams (ERDs)
4. Generate and execute database schema (DDLs)
5. Conceptual, logical and physical ERD support
6. Reverse engineer stored procedures to ERDs
7. Foreign key auto-naming
8. Define PK naming pattern
9. Define FK relationship naming pattern
10. Model primary key in object model by using the <<PK>> stereotype
11. Download database drivers automatically
12. Display database architecture in object-relational mapping (ORM) pane
13. Customizable SQL generation
14. Generate class diagrams from ERDs
15. Generate ERDs from class diagrams
16. Select target diagram when first synchronized between class diagram and ERD
17. Jump between ORM class and entity
18. Generate and reverse engineer database support for Oracle schema
19. Generate or reverse engineer user-defined database types
20. Database trigger and stored procedure modeling
21. Reverse engineer DDL models to ERD models
22. Database trigger and stored procedure generation and reversal
23. Unique and Index support for entities
24. Convert a normal association to ORM association
25. Support AbstractPersistable class for generating non ORM super class attributes

### Object-Relational Mapping

1. Java ORM code generation
2. .NET ORM code generation
3. PHP ORM code generation
4. Lazy collection fetching
5. Database view support

## Use Case Modeling

### Visual Modeling

1. Design Patterns support
2. Annotation with callout shapes
3. Annotation with freehand shapes
4. Organize model elements and diagrams using Model Explorer
5. Bookmark support
6. Handi-Selection tool
7. Customizable data types for use with multiple programming languages
8. Overview diagrams
9. Resource-centric interface
10. Model sharing
11. Cut, copy and paste
12. Copy diagrams as images for use in other applications
13. Undo and redo options
14. Mouse gestures
15. Reverse connector direction
16. Group creation support
17. Jump to feature for selecting a particular shape or model
18. Quick connect feature
19. Easy navigation to connected elements
20. Model commenting
21. Duplicate shapes and models
22. Selectable/non-selectable toggling for shapes
23. Diagram locking
24. Reference to any type of artifact
25. Advanced file and directory selector
26. Advanced tree support
27. Duplicate, move and reconnect connectors
28. Package headers for all types of diagrams
29. Tagged value display toggling for diagram elements
30. Sub-diagrams and reference indicators
31. Visio integration
32. Import Visio stencils
33. Rectilinear, round rectilinear, oblique, round oblique and curve connector styles
34. Space reclamation or elimination using Sweeper feature
35. Create shapes with user-defined initial sizes
36. Drag-and-drop creation of shapes using trees in diagrams
37. Auto-fit shape sizes
38. In-line editing
39. Spell checking
40. Visual alignment guides

## Use Case Modeling

### Visual Modeling

41. Numerous grid options
42. Diagram information display in diagrams
43. Jump to diagram feature
44. Drag-and-drop copying, moving and reordering of classes and entity members
45. Open view from model element
46. Diagram renaming boxes
47. Add folders as favorites
48. Display Undo/Redo action names
49. Inverse shape selection
50. Create new attribute with Enter key
51. Automatic reroute connector when overlapped with other shapes
52. Hide shapes or type of shapes on a diagram
53. Visualize related model element
54. Enforcing master view between model element and shape

### Model Element Nicknaming

1. Assign nicknames to model elements

### Model Transitor

1. Generate and link model elements
2. Trace the origin of model elements (model traceability)
3. Diagram Transitor
4. Navigate between operation and sequence diagram

### Style and Formatting

1. Customizable shape style and formatting
2. Image incorporation in diagrams
3. Stereotyped element appearance
4. Rich text documentation
5. Add rich text elements to diagrams
6. Identify candidate business process elements using textual analysis
7. Enrich model documentation with images
8. Save/load template for model documentation
9. Display stereotyped model element as image icon

### Team Collaboration with VP Teamwork Server

1. Concurrent and collaborative modeling with VP Teamwork Server
2. Import projects to VP Teamwork Server
3. View projects from VP Teamwork Server
4. Commit project changes to VP Teamwork Server
5. Update local project copy using VP Teamwork Server
6. Review past revisions using VP Teamwork Server

## Use Case Modeling

### Team Collaboration with VP Teamwork Server

7. Compare past revisions using VP Teamwork Server
8. Detect and resolve conflicts using VP Teamwork Server
9. Branch and tag projects using VP Teamwork Server
10. Merge branch changes using VP Teamwork Server
11. Export multiple revisions from VP Teamwork Server
12. Run VP Teamwork Server on common Java web servers
13. Element based revision history

### Team Collaboration with CVS Repository

1. Concurrent and collaborative modeling with CVS Repository
2. Import projects to CVS Repository
3. View projects from CVS Repository
4. Commit project changes to CVS Repository
5. Update local project copy using CVS Repository
6. Review past revisions using CVS Repository
7. Compare past revisions using CVS Repository
8. Detect and resolve conflicts using CVS Repository
9. Branch and tag projects using CVS Repository
10. Merge branch changes using CVS Repository
11. Export multiple revisions from CVS Repository
12. Element based revision history

### Team Collaboration with Subversion Repository

1. Concurrent and collaborative modeling with Subversion Repository
2. Import projects to Subversion Repository
3. View projects from Subversion Repository
4. Commit project changes to Subversion Repository
5. Update local project copy using Subversion Repository
6. Review past revisions using Subversion Repository
7. Compare past revisions using Subversion Repository
8. Detect and resolve conflicts using Subversion Repository
9. Branch and tag projects using Subversion Repository
10. Merge branch changes using Subversion Repository
11. Export multiple revisions from Subversion Repository
12. Element based revision history

### Team Collaboration with Perforce

1. Concurrent and collaborative modeling with Perforce Repository
2. Import projects to Perforce Repository
3. View projects from Perforce Repository
4. Commit project changes to Perforce Repository
5. Update local project copy using Perforce Repository
6. Review past revisions using Perforce Repository

## Use Case Modeling

### Team Collaboration with Perforce

7. Compare past revisions using Perforce Repository
8. Detect and resolve conflicts using Perforce Repository
9. Branch and tag projects using Perforce Repository
10. Merge branch changes using Perforce Repository
11. Export multiple revisions from Perforce Repository
12. Element based revision history

### Documentation Generation

1. PDF report generation
2. MS Word report generation
3. HTML report generation
4. Project publisher
5. Ad Hoc report creation
6. Intelligent element sorting during report generation

### Printing

1. Print multiple diagrams
2. Preview printable pages
3. Print clip marks
4. Page margin, size and orientation alteration support
5. Fit-to-pages option
6. Fit-to-ratio option
7. Customizable page header and footer
8. Project name and diagram name display in header or footer
9. Print with frame or border support
10. Toggle gradient color printing
11. Quick print support

### IDE Integrations

1. Activate full UML environment from your favorite IDE
2. Automatic code and model synchronization
3. Simple integration of any IDE
4. Import existing VP-UML project to IDE
5. Integration with Eclipse
6. Integration with NetBeans
7. Integration with IntelliJ IDEA
8. Integration with WebLogic Workshop
9. Integration with Borland JBuilder
10. Integration with Oracle JDeveloper
11. Multilingual support in IDE integration

## Use Case Modeling

### Reverse Engineering

1. Reverse engineer source code or executables to class diagrams using Instant Reverse feature
2. Reverse engineer Java source code, classes and .jar files
3. Reverse engineer C++ source code
4. Reverse engineer .NET .dll and .exe files
5. Reverse engineer CORBA IDL source code
6. Reverse engineer Ada 9x source code
7. Reverse engineer XML
8. Reverse engineer XML Schema
9. Reverse engineer databases with JDBC
10. Reverse engineer Hibernate mapping files
11. Reverse engineer PHP 5.0 source code
12. Reverse engineer Python
13. On-demand Java reverse engineering
14. Template parameter support

### Code Generation

1. Instantly generate source code from class diagrams
2. Generate Java source code
3. Generate C# source code
4. Generate VB.NET source code
5. Generate PHP 5.0 source code
6. Generate Object Definition Language source code
7. Generate Flash ActionScript 3.0 source code
8. Generate IDL source code
9. Generate C++ source code
10. Generate Delphi source code
11. Generate Perl source code
12. Generate XML Schema source code
13. Generate Python source code
14. Generate Objective-C source code
15. Generate Ada source code
16. Generate Ruby source code
17. Template parameter support
18. Template support for generating source code with generic constructs
19. Customizable source code generation

### State Machine Diagram Code Generation

1. Generate Java source code from state diagram
2. Generate C++ source code from state diagram
3. Generate C# source code from state diagram
4. Generate VB.NET source code from state diagram

## Use Case Modeling

### Java Round-Trip Engineering

1. Reverse engineer Java source code to class diagram
2. Update Java source code based on class diagram

### Shape Editor

1. Design arbitrary shapes
2. Import SVG shapes
3. Incorporate different shapes into UML diagram
4. Organize shapes by gallery, category and stencil
5. Advanced shape design capabilities

### Layout Facilities

1. Automatic diagram layouts
2. Shape alignment and centering
3. Uniform shape width and height maintenance
4. Automatic shape distribution

### Interoperability

1. Import Telelogic Modeler project files
2. Import and export EMF based UML2 model
3. Command-line operations
4. Import and export XMI 1.0, 1.2 and 2.1
5. Import and export XML
6. Import and export VP project file format
7. Import and export use case model to MS Word
8. Import Rational Rose project files
9. Import ERwin data modeler project files
10. Generate BPEL code for Oracle workflow engine
11. Generate BPEL code for JBoss workflow engine
12. Export diagrams as JPG, PNG, SVG and EMF image files
13. Export diagrams as PDFs
14. Slice exported diagrams into smaller segments
15. Import Rational Software Architect files
16. Copy diagram elements as XML
17. Import Visio drawings into Visual Paradigm
18. Import and export Microsoft Excel file for class diagram
19. Import and export Microsoft Excel file for state machine diagram
20. Import and export Microsoft Excel file for component diagram
21. Import and export Microsoft Excel file for deployment diagram
22. Import and export Microsoft Excel file for package diagram
23. Import and export Microsoft Excel file for requirement diagram
24. Import and export Microsoft Excel file for entity relationship diagram
25. Import and export Microsoft Excel file for orm diagram
26. Import and export Microsoft Excel file for business process diagram

## Use Case Modeling

### Intuitive User Interface

1. Group diagrams by category
2. Advanced property pane
3. Dockable user interface windows.
4. New project generation using predefined templates
5. Easy-to-use “New Diagram” dialog
6. Flexible zooming
7. Palette-style toolbar
8. Collapsible toolbar
9. Display tool names for toolbar buttons
10. Expand grouped toolbar buttons
11. Display various diagram categories in toolbar
12. Numerous looks-and-feels
13. Import user preferences from existing workspaces
14. Multilingual support
15. Searchable options

### Automatic Updates

1. Automatic online updating
2. Maintain local update server with update synchronizer

### Open Architecture

1. Plug-in support (Java)
2. VP model and XML interaction

Visual Paradigm International

Tel: +852 2744 8722

Fax: +852 2744 6722

[sales@visual-paradigm.com](mailto:sales@visual-paradigm.com)

[www.visual-paradigm.com](http://www.visual-paradigm.com)

© 2008 Visual Paradigm International All rights reserved.