

Applying Requirements Management with Use Cases

The Practice and Discipline of
Requirements Management

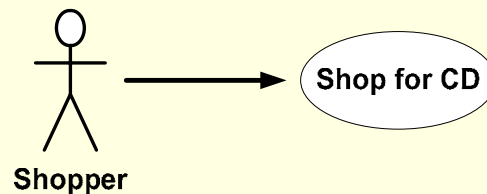
May 28, 2006

Use Case Defined

A use case describes a sequence of actions a system performs that yields an observable **result of value** to a particular actor

Use Cases tie the different UML models together by driving Architecture, Design, and Testing

Use Cases are Shown in UML Diagrams



Use Cases are Described in Text

Use Case Specification: **UC5 Shop For CD**

1. Brief Description
UC5.1. In the past year the Shopper can access information on all available CDs and find the music to purchase, compare prices, get information. The system must not require users assistance in searching (links) (see for details.)

2. Actors
2.1. Primary Actors
2.1.1. Shopper

3. Flow of Events
3.1. Basic Flow
3.1.1. **UC5.1.BASIC**
The user will begin with the Shopper Access to the CD's

3.1.2. **UC5.1.1. HEURISTIC SUGGESTION**
The system presents:
1. A recommended selection of the list
2. Search criteria entered
3. A list of top artists
The Shopper Access to the CD's

3.1.3. **UC5.1.2. OPTIONS**
The system presents options to:
1. View a list of all available CDs
2. Search for CD's
The Shopper Access to the CD's

3.1.4. **UC5.1.3. INFORMATION ABOUT MUSIC**
The system presents the available titles in alphabetical order by Catalogue. Each line contains the title of the album, the year released, the performing artist and the price.

3.1.5. **UC5.1.4. VIEW CD LIST**
The Shopper Access to CD is listed in the details. The system displays the following information:
1. Name of the CD
2. Price
3. Performer and Artists
4. Available CD's information
5. Name of CD's

3.1.6. **UC5.1.5. CHOOSE ITEM**
The Shopper Access to the CD is shown when the user clicks on the Shopper Access to the CD. The system displays the following information:

Use Cases are Requirements

Use Cases are about specifying “Functional Requirements”

FURPS

- Functionality
- Usability
- Reliability
- Performance
- Supportability

Design Constraints

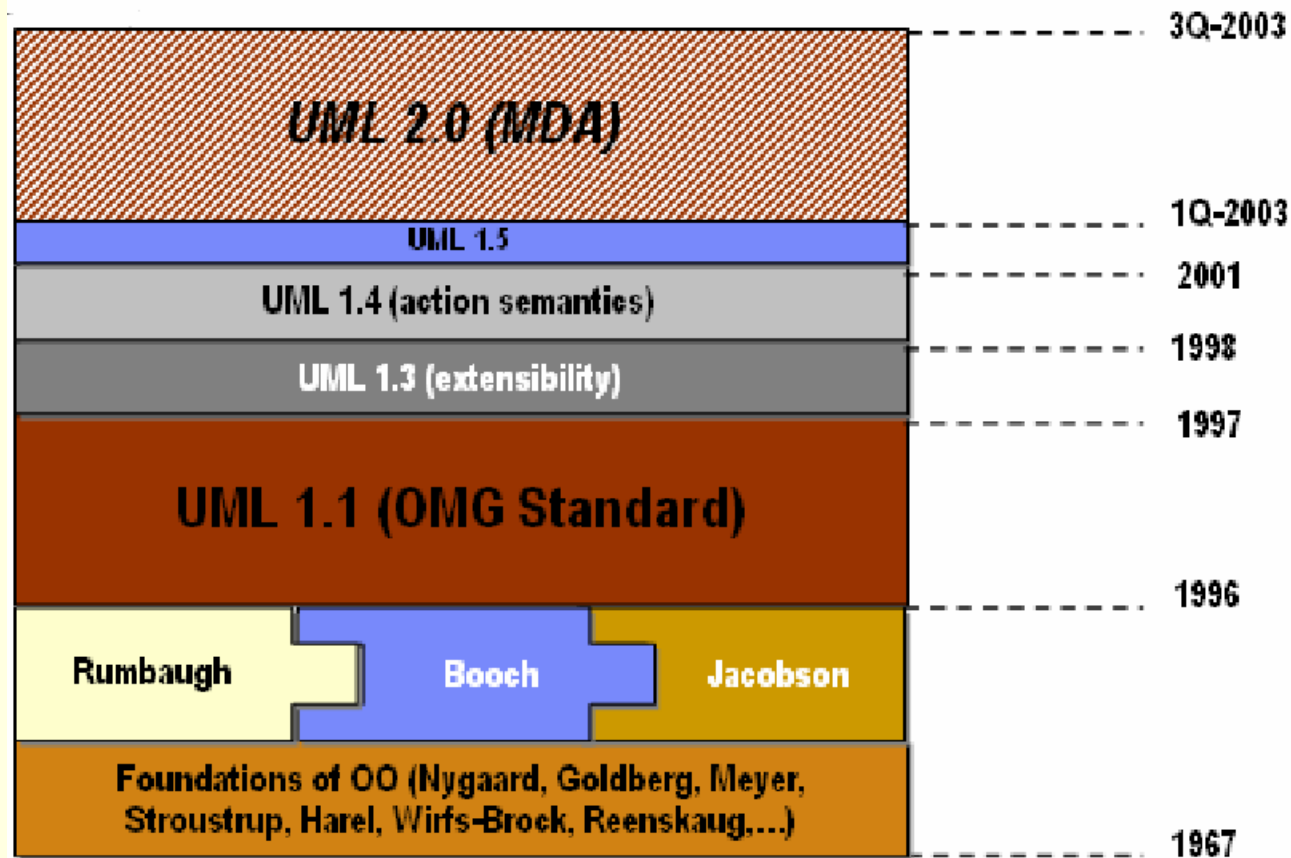
Environments
Compatibility
Application Standards
User Interface

Legal and Regulatory Requirements

- Federal Communication Commission
- Food and Drug Administration
- Department of Defense
- OSHA

Use Case process drives out Supplementary Requirements

Use Cases are Part of the Unified Modeling Language (UML)



Theory of Value

Use Cases Create Value

Economics

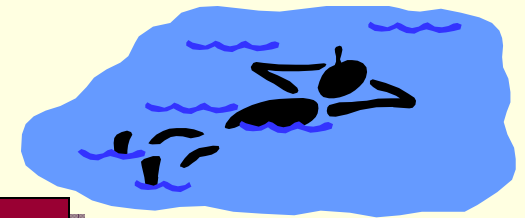
- the theories within economics that explain the worth of goods and services
- Intrinsic (objective) theories
 - the value of goods and services is a property built into the item itself
- Subjective theories
 - the value of goods and services is a judgment call of the person wanting those goods and services

“meaning-making”

- Items that are not consumption (purchased) and...
- are not production (creating and/or modifying something intended for sale)
 - cooking a meal, extinguishing a fire

Paradox of Value

- Diamond-water paradox
 - Diamonds have **economic** value
 - Water has **use** value
- Which is of greater value?

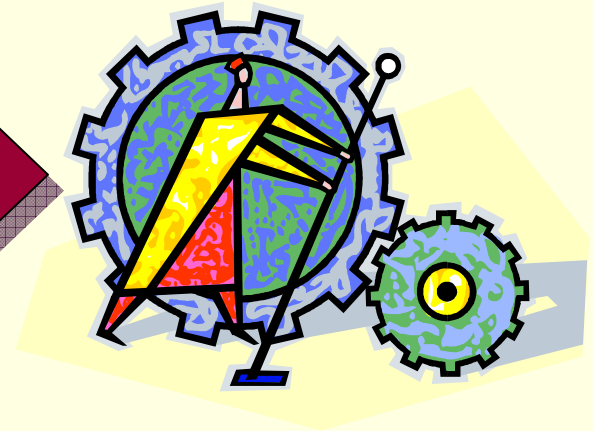


The Actor Defines the Value

Project Software Requirements Specification

Details of how these features are going to be implemented in the solution

Goal: Move from Vision to SRS



- Authored by the Requirements Analyst or Business Analyst
- Is Related to the Vision document

Software Requirements Specification (SRS)

Acts as the primary “Package” containing all requirements

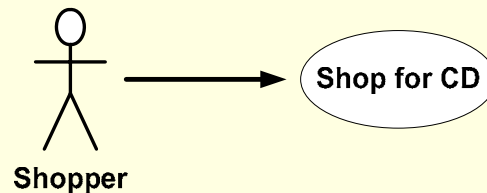
- Functions to collect and organize project requirements
 - Use Cases
 - Use Case Models
 - Supplementary Requirements Specifications
- Many different arrangements of an SRS are possible

Use Case Defined

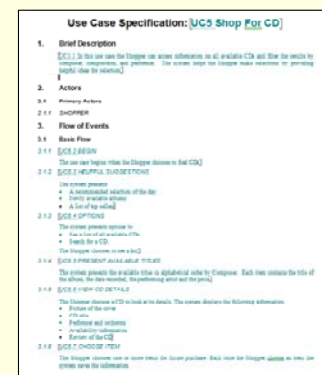
A use case describes a sequence of actions a system performs that yields an observable **result of value** to a particular actor

Use Cases tie the different UML models together by driving Architecture, Design, and Testing

Use Cases are Shown in UML Diagrams



Use Cases are Described in Text



Alternative SRS Styles

Traditional SRS

- Captures Functional Requirements and not use cases
- Captures all requirements in a single document
- Captures non-functional requirements in one document

Use Case SRS

- Captures use case model surveys and use case reports
- Requirements may be in in one or many use cases
- Captures supplementary specifications in one or many use cases

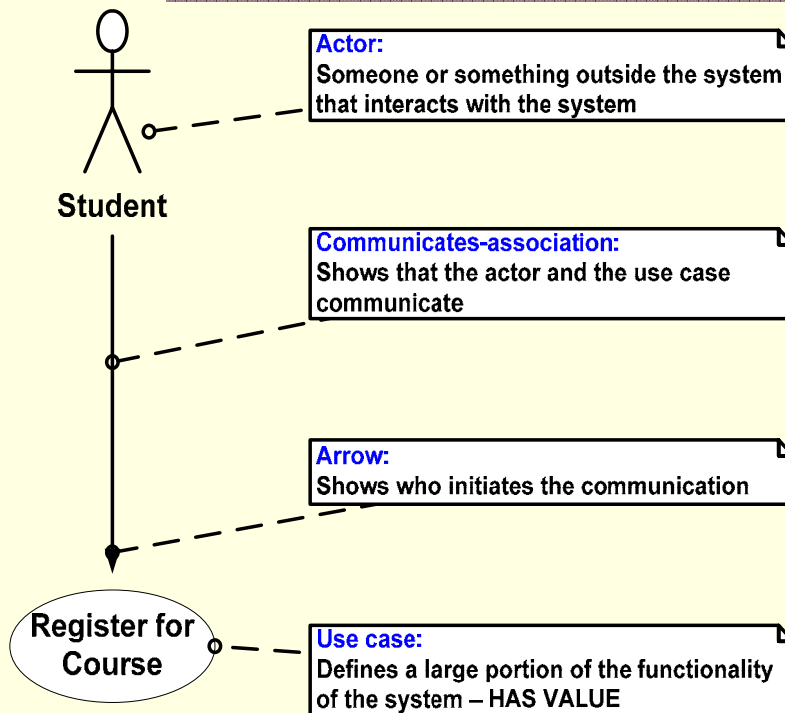
What is UML?

The UML is the standard language for specifying, visualizing, constructing, and documenting all the artifacts of a software system.”

- *Terry Quatrani, UML Evangelist*

UML Symbols: Actors and Associations

Use Case diagrams are not about data flow or control flow!

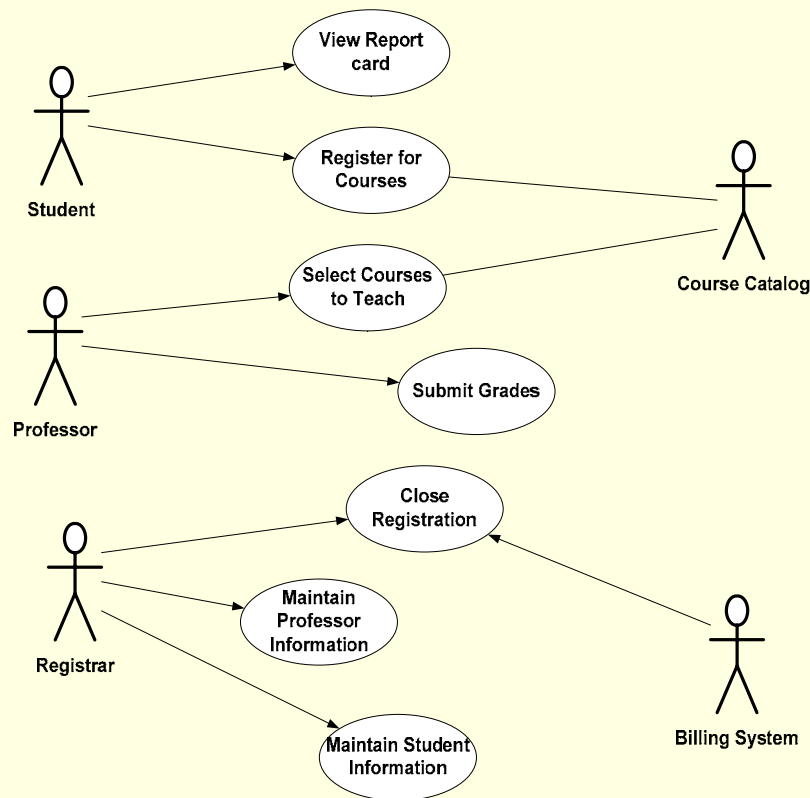


Example of Actors as Human Beings

Use cases are also good for real-time systems (non-human interaction)

Use Case Diagrams

Defines “What” value the system provides



Provides a view of the entire system

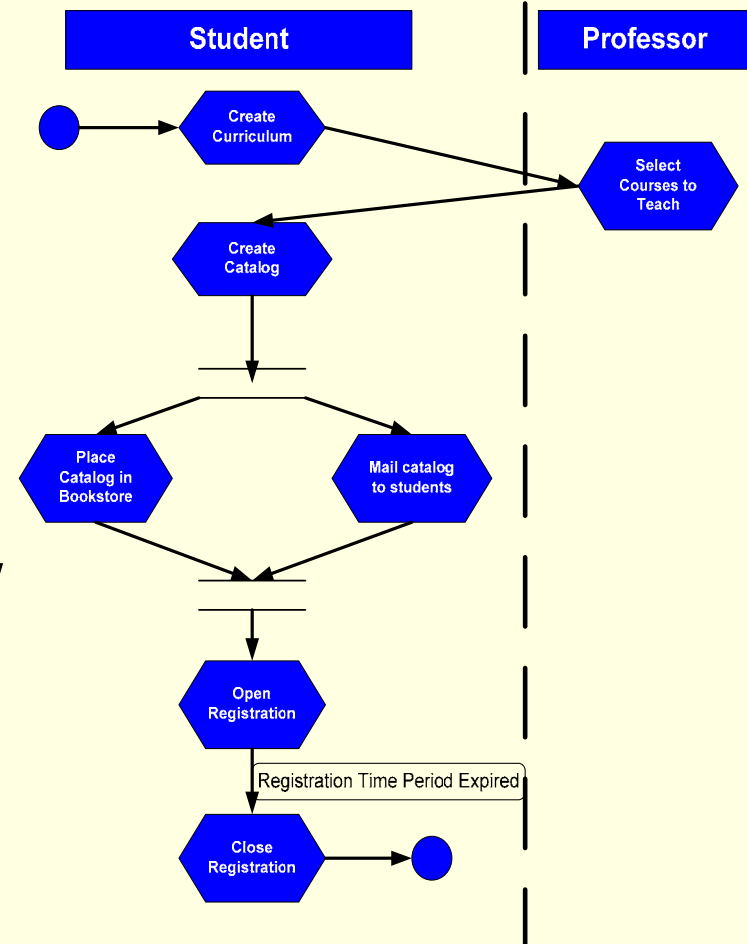
Sets boundaries: all actors are “outside the system”

Represents both human and non-human actors

Note: Actors do not communicate with Actors, Use Cases do not communicate with Use Cases

UML - Activity Diagrams

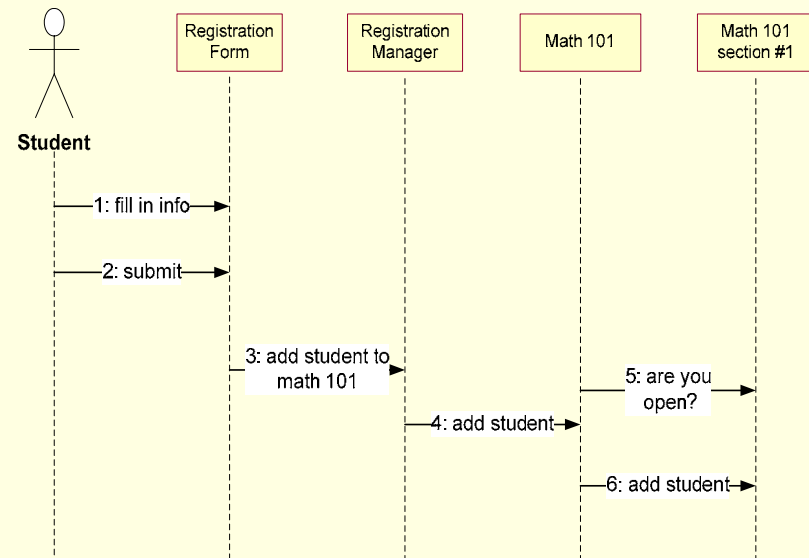
- Show flow of control
- Used during analysis and design process to show business workflow
- Shows how things flow between use cases
- Swim lanes show ownership



UML - Sequence Diagrams

Defines “How” the system provides value

- Shows object interactions arranged in a time sequence
- The flow of events determines what objects and interactions are needed to accomplish the functionality specified by the flow of events
- **Is the use case realization**



Use Case Textural Standards

Today we will focus on RUP Standards

There is no Standard for writing Use case specifications

The UML does not specify how the text of a use case should be structured, organized, or written

How Many Use Cases is too Many?

Key is to remember the definition

A use case describes a sequence of actions a system performs that yields an observable **result of value** to a particular actor

- Understanding the “value” helps to define the level of granularity
- Biggest mistake is to write too many small use cases

4 Requirements for a Use Case

1. Must provide value to the stakeholder
 - Goal Oriented
2. Must be a complete narrative describing how the value is provided
 - Must have Main and Alternative flows
 - Power of a use case is that it is “natural language”
3. Must stand alone
 - No sequencing of use cases
 - Only lines between use cases are includes, excludes, generalization (advanced topics)
4. Must not describe design
 - “What” not “How”

Use Cases vs. Declarative Statements

Declarative

- The system shall provide a secure login.
- The system shall provide a list of class offerings for the current semester.
- The system shall only allow registration for courses where the prerequisites are fulfilled.
- The system shall provide a confirmation number when the schedule is submitted.

- Declarative “shall” statements
- Small perspective
- System orientation

Use Case

1. The student enters a student ID and password and the system validates the student.
2. The system presents a list of course offerings. The student chooses up to four...
3. The system displays the functions available to the student: create, modify, delete.
4. The system validates the courses selected and displays a confirmation number...

- Broad perspective
- Goal oriented
- Actor (user) focused

The Contents of a Use Case

Use Case Name

1. Brief Description
2. Actors
3. Flow of Events
 - i. Main Flow
 - ii. Alternative Flows
 - i. Alternative flow 1
 - ii. Alternative flow 2
4. Special Requirements
 - i. Business Rules
 - ii. Usability Requirements
 - iii. Data Definitions
5. Pre-conditions
6. Post-conditions

Special Requirements

- Non-functional requirements specific to a use case
- Keeps it close to the use case

Pre-condition

- Needs to be “true” for the use case to start
- Usually some “state” of the system

Post-conditions

- The required state of the system once the use case ends
or...
- Some action the system must take once the use case ends
- Avoid conditions that are about the state of the “world”

The Contents of a Use Case

Use Case Name

1. Brief Description
2. Actors
3. Flow of Events
 - i. Main Flow
 - ii. Alternative Flows
 - i. Alternative flow 1
 - ii. Alternative flow 2
4. Special Requirements
 - i. Business Rules
 - ii. Usability Requirements
 - iii. Data Definitions
5. Pre-conditions
6. Post-conditions

Main Flow

- “Happy Path”
- “Path of least resistance”
- Should be only one Main Flow

Alternative Flow

- Regular “Variance”
- Odd cases
- Exceptional (Error) flows

Common Mistakes

- Multiple “Main Flows”
Difficult to see the goal being accomplished
- No “Alternative Flows”
Not really a use case
Exception may be a draft or early use cases

Use Case Styles

“Style” is how you structure the text of the use case

- Use Case Specifications can be considered “structured text”
- There are many different styles to choose from
- For a given project (or organization) it is vital to choose and be consistent with one style
 - For consistency
 - For readability
 - For usability by the development team

Use Case Style Considerations

Documented in the Use Case Modeling Guidelines

- Do steps in the “Flows” have numbers or titles or both?
- Do “Alternative Flows” have numbers or titles or both?
- Does the “Main Flow” reference other flows or not?
- How to you reference one part of a use case from another?
- Can “Flows” have embedded flows?
- How do “Alternative flows” tell what happens when they are done?

RUP Style Main (Basic) Flow of Events

Main flow shows the actor succeeding in his/her goal

Use Case Specification: Register for Courses

Brief Description

This use case allows a Student to register for course offerings in the current semester. The student can also modify or delete course selections if changes are made within the add/drop period at the beginning of the semester. The Course Catalog System provides a list of all the course offerings for the current semester.

Actors

1. *Primary Actor – Student*
2. *Secondary Actor – Course Catalog System*

Flow of Events

1. Basic Flow

1.1 LOGON

This use case starts when a student accesses the Course Registration System. The student enters a student ID and password and the system validates the student.

1.2 CREATE SCHEDULE

The system displays the functions available to the student. These functions are: Create a Schedule, Modify a Schedule, and Delete a Schedule. The student selects "Create a Schedule."

1.3 SELECT COURSES

The system retrieves a list of available course offerings from the Course Catalog System and displays the list to the student. The Student selects up to 4 primary course offerings and 2 alternative course offerings from the list of available offerings. The student can add and delete courses as desired until choosing to submit the schedule.

1.4 SUBMIT SCHEDULE

The student indicates that the schedule is complete. The system validates the courses selected and displays the schedule to the student. The system displays the confirmation number for the schedule. The system saves the student's schedule information. The use case ends.

Structure the flow into steps

Number and title each step

Describe steps in 1 to 3 sentences

Don't refer to alternative flows in the main flow

Confidential

©J Rivera & Associates, Inc
2005

1

RUP Style Alternative Flows of Events

Alternative flows are flat

The can have steps

They have names

2. ALTERNATIVE FLOWS

2.1 MODIFY A SCHEDULE

AT BF CREATE SCHEDULE, the Student already has a schedule that has been saved; the system retrieves and displays the Student's current schedule (e.g. the schedule for the current semester) and allows him/her to use it as a starting point. The use case resumes at BF SELECT COURSES.

2.2 DELETE A SCHEDULE

AT BF CREATE SCHEDULE, the Student has an existing schedule and chooses to delete it. The System retrieves and displays the Student's current schedule. The System prompts the Student to verify the deletion. The Student verifies the deletion. The System deletes the schedule. The use case ends.

2.3 UNIDENTIFIED STUDENT

AT BF LOG ON, the System determines that the student is not valid, an error message is displayed and the use case ends.

2.4 QUIT

The Course Registration System allows the Student to quit at any time during the use case. The Student chooses not to save any partial schedule information. The use case ends.

2.5 QUIT AND SAVE

The Student chooses to quit creating a schedule and chooses to save a partial schedule before quitting. All courses that are not marked as "enrolled in" are marked as "selected" in the schedule. The System saves the schedule. The use case ends.

2.6 CANNOT ENROLL

AT BF SUBMIT SCHEDULE, the System determines that the prerequisites for a selected course are not satisfied, or that the course is full, or that there are schedule conflicts, the System will not enroll the Student in the course. The System displays a message to the Student and the use case continues at BF SELECT COURSES.

2.7 COURSE CATALOG UNAVAILABLE

AT BF SELECT COURSES, the System determines that the Course Catalog is system is not available. The System displays an error message and the use case ends.

2.8 REGISTRATION CLOSED

AT BF LOG ON, the System determines that registration is closed; the system indicates that the user can no longer select courses and the use case ends.

1) Start:

Indicate the starting point. May be basic flow or alternative flow

2) Cause:

Say what causes the flow to start

3) Effect:

Say what happens as a result of the cause

4) End:

Say where the flow resumes or if it ends

Confidential

©J Rivera & Associates, Inc
2005

2

Alternative Use Case Styles

Research Options

- Tagged style
(Bittner and Spence)
Good
Introduces Subflows
and extension points
- “A Survey of Approaches
for Describing and
formalizing Use Cases”
Russell R. Hurlbut
describes over 30
approaches to use
case modeling

Use Case Guidelines

- Make a conscience
decision about style
- Document the style in the
Use Case modeling
Guidelines
Is a RUP Artifact
Helps enforcement
- Enforce the chosen style

More Information

- *For More Information Contact*
 - *James Rivera, PMP*
 - *J Rivera & Associates, Inc*
 - *james@rivera-associates.com*
 - *info@rivera-associates.com*
 - *www.rivera-associates.com*