UML: From Use Case to User Acceptance

NAACCR 2002 Annual Meeting
Cancer Informatics Workshop
Sunday, June 9, 2002
Toronto, Canada
Session Outline

- The Use Case Model
- Specifying Activities
- What are Classes?
- How UML Becomes Software
- Keys to Success
- The Value of Iteration
- Interactive UML Group Activity
  - Developing a Model for Electronic Cancer Finding (E-Path)

30 Minutes

30-60 Minutes
Contact Information

Jack K. Golabek, P.Eng
Vice-President Engineering

Artificial Intelligence in Medicine Inc.
2 Berkeley Street, Suite 403
Toronto, Ontario M5A 2W3
416-594-9393 ext. 226

jgolabek@aim.on.ca
www.aim.on.ca
The Use Case Model

- Specifies all types of users
- Specifies all types of user interactions
- Describes the results of user interactions

The Use Case Model tells us who the users are and how they interact with the system.
The Essence of a Use Case

*What is the system supposed to do for each type of user?*

- Identifies a user interaction
- Specifies any pre-conditions
- Specifies the results/products of the interaction
- Lists the functional requirements of the interaction
- Lists policy and business rules (if any)
Example Use Case Model

Tracking, follow-up and prevention of cervical cancer.

Source:
AIM Inc.
ISIS-CSP Software Specifications
Copyright © 1996-2002
Activity Diagrams

- Describe a sequence of actions (workflow)
- Describe responsibility (who does what)
- Describe conditions and branches

Activity diagrams tell us how things are done and according to what rules.
Example Activity Diagram

Automated Record Linkage

Source:
AIM Inc.
ISIS-Registrar (Enterprise)
Copyright © 1998 - 2002
What is a Class?

• A class is an abstraction of something
• A class is described by attributes
• A class can perform operations
• An instance of a class is a specific thing

Class diagrams describe system components, what they’re for, and what they do.
Example Class Diagram

Histology Entity Relationships

Source:
AIM Inc.
E-Path Design
Copyright © 1998 - 2002
Example Class Diagram

Electronic Transmission of Pathology Reports

Source:
AIM Inc.
TransMed EDI Design Documentation
Copyright © 1996-2002
How UML Becomes Software

Use Cases = Users & Interactions
Classes = Components
Activities = Logic & Rules

User Interfaces
Architecture
Data & Code

“The System”

Very simplified. The process is actually more complex and usually iterative.
Use Case Driven Development

Source: The Unified Software Development Process: Jacobsen, Booch, Rumbaugh, 1999
Keys to Success

• Specifications & analysis come first.
• Design before you build.
• Test early and often.
• Document all components.
• Acceptance doesn’t mean the end.
The Value of Iteration

• Recognize that not everything can be specified in advance.
• New ideas emerge as systems are developed.
• Anything that can go wrong, will!
• Iteration identifies and mitigates risk.
• “Iterative” is not the same as “endless”.

The Unified Process

• “Unified” means client and vendor are working together using a common framework*.
• Focused on value (use case driven)
• Architecture centric (reusability, scalability)
• Iterative (mitigate risk)

*Unification actually refers to the consolidation of the original object oriented design methods developed by Jacobson, Rumbaugh and Booch, but I propose that what UML also accomplishes is a unification between client and vendor in the understanding of a system.
Hands-On Exercise

Statement of Objectives
Developing a Use Case Model
Writing Use Case Specifications
Writing User Acceptance Criteria
Objective

A cancer registry wishes to improve the process of cancer case finding and data consolidation in its jurisdiction. The metrics to be used in assessing success are:

– Timeliness of Case Finding
– Completeness and Quality of Data
– Operating Costs
The Use Case Model

- Who are the actors?
- What work is performed as-is?
- Identify problems/issues with the as-is model.
- What can we change? Improve?
- Develop the to-be model
Writing Specifications

- Use Case ID
- Context
- Actors
- Work/Process Description
- List of Functional Requirements
- List of Policies/Rules
User Acceptance

- Use Case ID
- Implementation Notes
- Fulfillment of Functional Requirements
- Policy/Rule Validation
- Acceptance Status
The End

Thank you for your attention.