SECP 2005

Teaching Object-Oriented Software Development through Course Projects: a Success Story

Carleton UNIVERSITY
Canada's Capital University

Presented by:
Juan Pablo Zamora Zapata

2nd Software Engineering Course Project (SECP 2005) Workshop. CASCON 2005. Toronto, ON, Canada. October 2005

Carleton UNIVERSITY
Canada's Capital University

Outline

- Introduction
- Scope of the Projects
- Project Descriptions
- Interaction with the "Client"
- Traceability Model
- Tutorials
- Workload and Deliverables
- Group Dynamics
- Sample Project
- Conclusion

Introduction

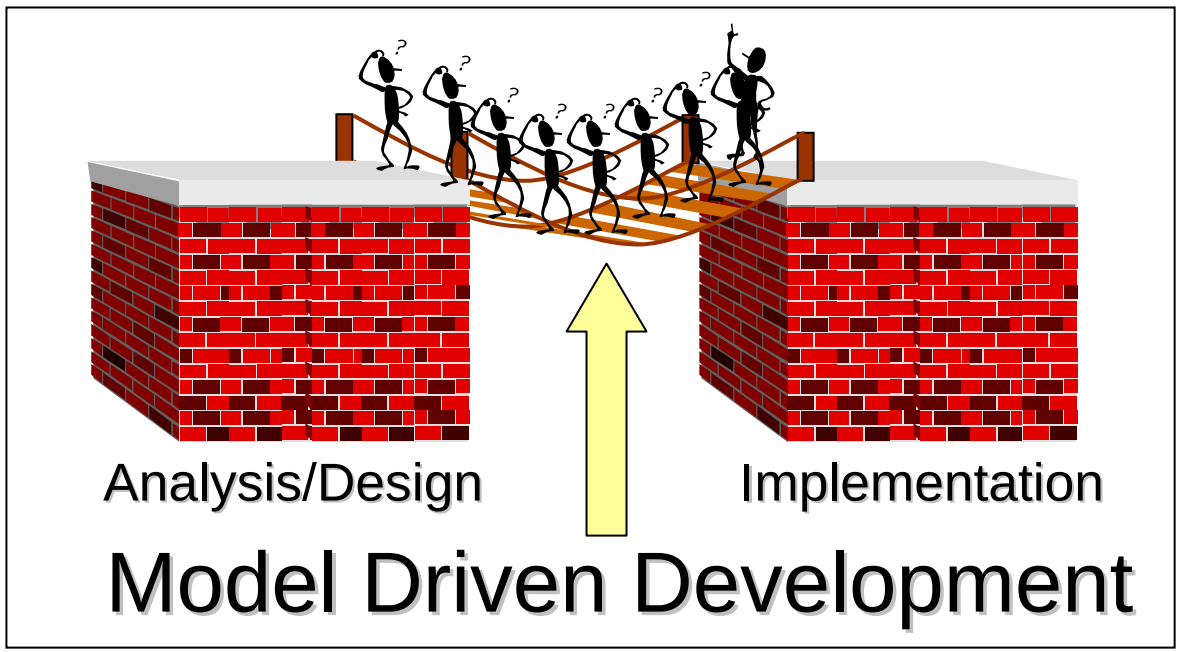
Carleton UNIVERSITY
Canada's Capital University

Characteristics of our Course Projects:

- Fully operational distributed system
- Different technologies and programming languages are used
- Team-based development
- Client-server applications
- Model Driven Development environment
- Strong emphasis on:
  - Modeling
  - Documentation
  - Traceability

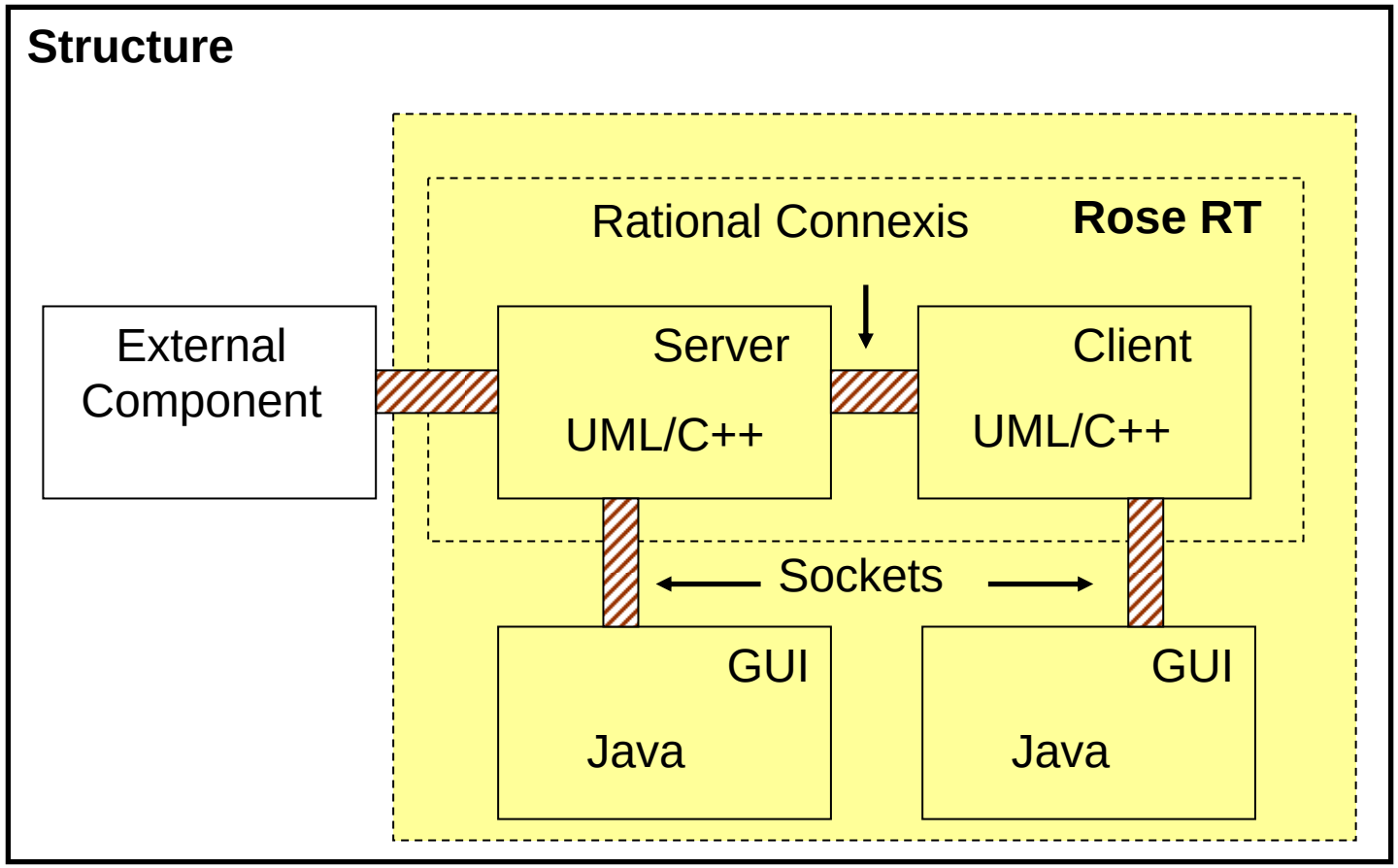
Why Model Driven Development?

- We aim for quality on models and documentation
- Models in-sync with implementation
- Code embedded into models
- Models as the foundation for maintainability



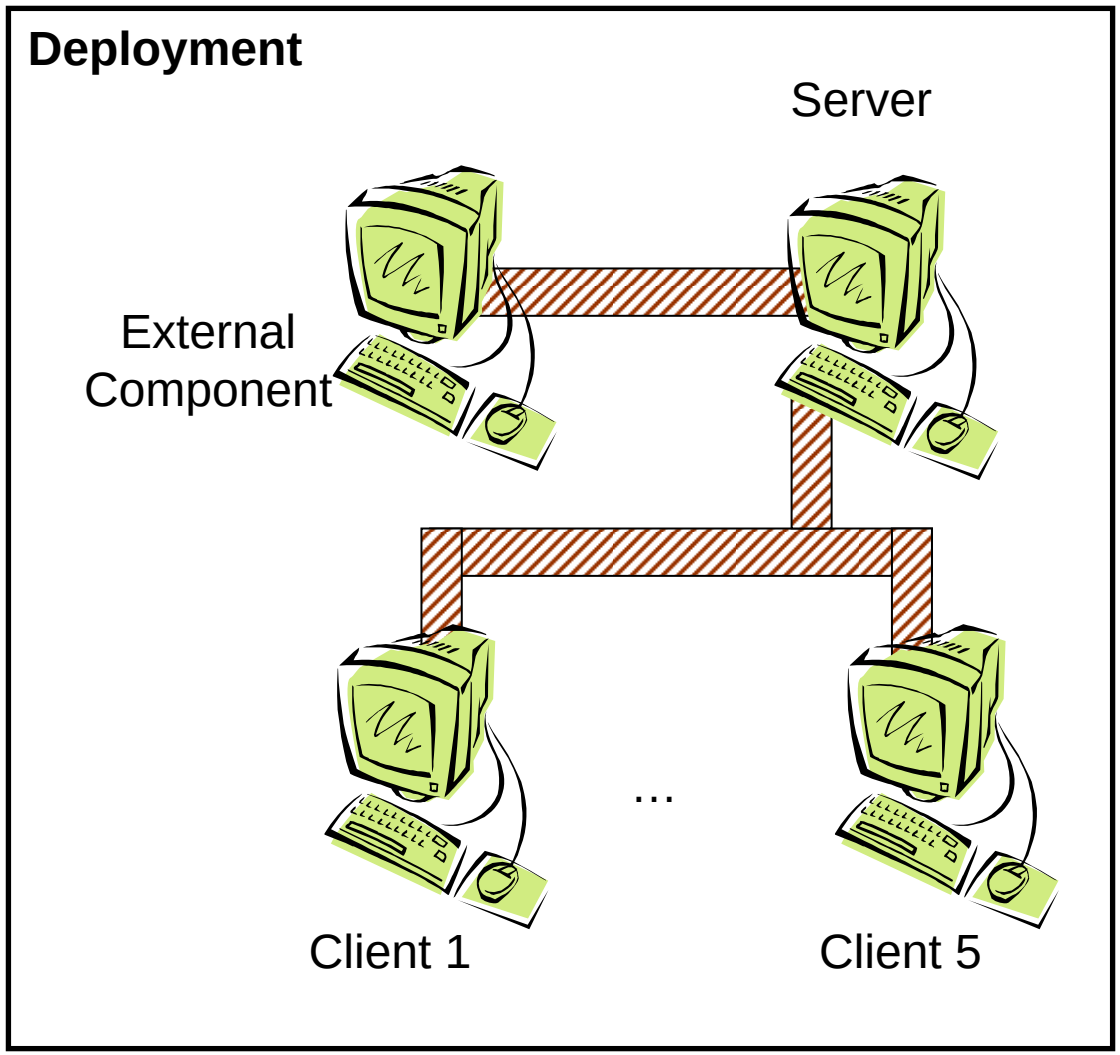
Carleton UNIVERSITY
Canada's Capital University

Scope of Projects



Carleton UNIVERSITY
Canada's Capital University

Scope of Projects

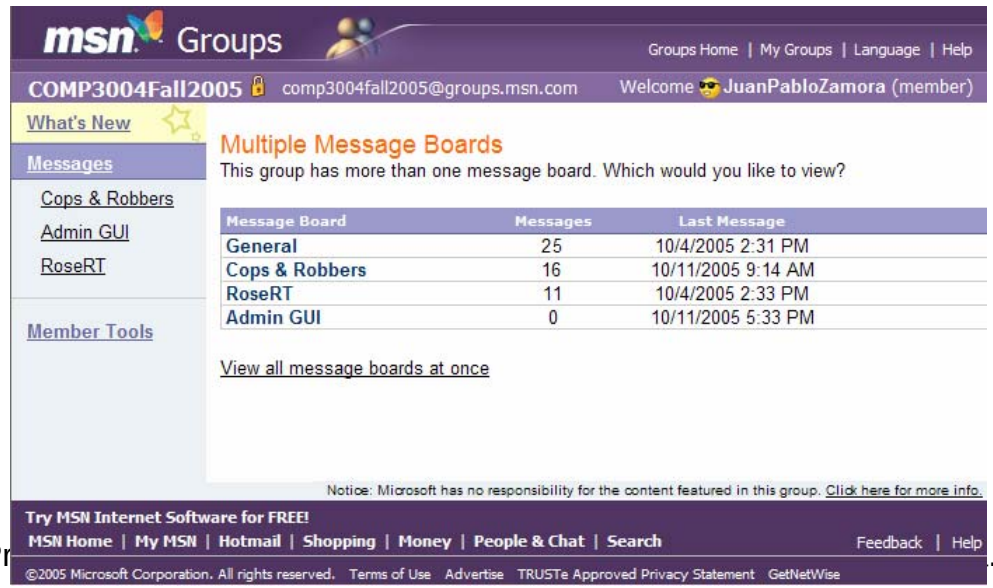


Project Descriptions

- General idea or topic
- Students are familiar with the domain
- Deliberately vague in its details:
  - Students have enough to start
  - Gathering requirements is a discovery process between software analyst and client
  - Being vague allows the instruction team for easy manipulation and introduction of changes
  - The project can be expanded or simplified depending on student progress

Meeting with the Client

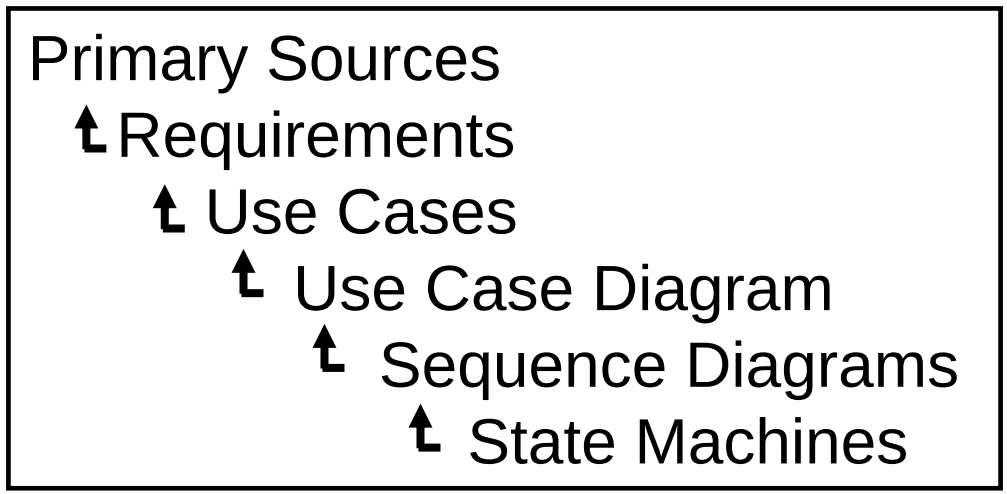
- Online discussion group
- Postings usually translate into new/modified requirements
- Maximizes TA resources :
  - Consistency among the instruction team
  - No duplication of Answers
  - No coordination between TAs is required



Carleton UNIVERSITY
Canada's Capital University

Traceability Model

- From implementation back to requirements
- From requirements forward to implementation
- Basis for future maintenance



Carleton UNIVERSITY
Canada's Capital University

Tutorials

- Focused on practicing recently acquired knowledge
- Targeted topics:
  - Capsule communication using Protocols, Ports and Signals (LightSystem)
  - Good modeling practices (Speed Machine)
  - User Interaction (Java GUIs)
  - Connexis communication (Distribution)

Workload and Deliverables

Document Proposed Schedule

COMP 3004 Object-Oriented Software Engineering		Suggested Work time		Winter Break		March 12 Last day to withdraw from the course							
Document Follow Up Schedule													
		Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13
		5-Jan	12-Jan	19-Jan	26-Jan	2-Feb	9-Feb	16-Feb	23-Feb	1-Mar	8-Mar	15-Mar	22-Mar	29-Mar
Document Sections														
	Introduction													
	Short introduction to section													
	Objective													
	Content													
	RRRT model description													
	Relationship between RRRT and doc													
	Description of System													
	Short introduction to section													
	Short system description													
	Scope of project													
	Assumptions													
	Short introduction to section includes explanation of notation													
	Quality of assumptions													
	Unique identifiers													
	Assumptions are justified when necessary													
	Requirements													
	Short introduction to section includes explanation of notation													
	Functional requirements													
	Non-functional requirements													
	Unique identifiers													
	Facility to information (how easy is it to find the necessary requirements)													
	Use Cases													
	Short introduction to section includes explanation of notation													
	Use case diagram													
	Facility to information (how easy is it to find a use case; i.e. Do they provide a table to locate them)													
	Are the use cases complete enough to cover the required functionality of the system													
	Are the use case steps at the appropriate level of granularity													
	Are the use cases clear and easy to understand													
	Are the individual use case descriptions complete with sufficient information (a la STD template)													
	Design Decisions													
	Short introduction to section													
	Quality of decisions													
	Justification provided for decisions													
	Patterns referenced where appropriate													
Traceability														
	Requirements to Project description, discussion and assumptions													
	Use case descriptions to requirements and assumptions													
Overall Quality														
	Document quality, spelling, grammar, etc.													

Carleton UNIVERSITY
Canada's Capital University

Workload and Deliverables

RoseRT Models Proposed Schedule

COMP 3004 Object-Oriented Software Engineering	Suggested Work time				Winter Break			March 12 Last day to withdraw from the course					
RT Models Follow Up Schedule													
	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13
	5-Jan	12-Jan	19-Jan	26-Jan	2-Feb	9-Feb	16-Feb	23-Feb	1-Mar	8-Mar	15-Mar	22-Mar	29-Mar
RT Models													
Diagrams													
Use case diagram													
Use case diagram contains use case descriptions													
Sequence diagrams													
Sequence diagrams and alternatives easy to relate to one another													
Structural model													
Structural model is clean and easy to read with appropriate names etc.													
Ports and protocols are logical in their usage													
Behavioural model													
Behavioural model is clean and easy to read with appropriate names etc.													
Behavioural model only uses single state when appropriate and justified													
GUI													
Simple and intuitive													
Clean													
Separate gui...													
Scenario execution													
Scenario 1													
Scenario 2													
Scenario 3													
Scenario 4													
Scenario 5													
Scenario 6													
Scenario 7													
Scenario 8													
Requirements to check													
Requirement 1													
Requirement 2													
Requirement 3													
Requirement 4													
Requirement 5													
Traceability													
Sequence diagrams to Use case descriptions													
Statecharts to sequence diagram													
Justification provided for decisions													
Patterns referenced where appropriate													
Overall Quality													
Model quality, naming, etc.													

Carleton UNIVERSITY
Canada's Capital University

Group Dynamics

- Social component of the course
- Multiple differences among students:
  - Cultural backgrounds
  - Working habits
  - Preference for programming languages
- For most students, it is the first time they are asked to work in groups of 4 people
- Teaching staff ‘helps’ students to create their own teams
- Peer evaluation

Carleton UNIVERSITY
Canada’s Capital University

SECP 2005

Sample Project

Carleton UNIVERSITY
Canada's Capital University

Conclusion

- Valuable skills:
  - Software development
  - Ability to accomplish a large project while working in a group
  - Time management
- Introduced to a model driven development methodology and UML
- Gain technical experience with the Rose-RT tool

- Our course exposes students to as much of a real world examples as can be accomplished in the context of a thirteen week undergraduate course

Questions