Component Oriented Software Development

A new Approach and a Case Study

Ali H. Doğru
Middle East Technical University

Contents

Introduction

- Unified Modeling Language: UML (OO)
- Component-Oriented Software Engineering Modeling Language: COSEML (CO)
- A case study including 20 student projects
- 3 group of metrics to evaluate process and product

UML

- Abbreviation for Unified Modeling Language
- Object Oriented Modeling, heavily based on classes, inheritance, and messages
- Defines 9 types of diagrams to represent various modeling viewpoints:
  - Use Case Diagrams
  - Class Diagrams
  - Sequence Diagrams
  - Collaboration Diagrams
  - Object Diagrams
  - Activity Diagrams
  - Statechart Diagrams
  - Component Diagrams
  - Deployment Diagrams

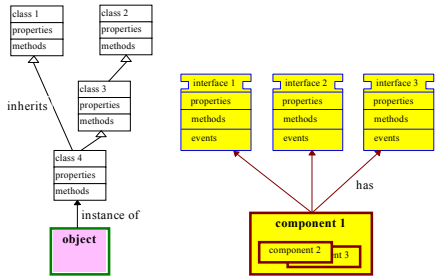
COSEML

- Supports Component Oriented Development
- Primary view is the structural decomposition
- Representation of abstractions as well as implementation-level components
- Static (composition) and dynamic modeling (message) links
- Decomposition view supported with UML syntax
- Yet experimental

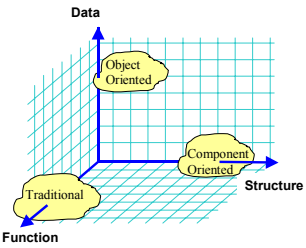
Component-Based versus Component-Oriented

- The system that are not fully object-oriented but that consist of objects are called object-based.
- Similarly, systems that are not fully component-oriented but consist of components are called component-based.

Objects Compared to Components



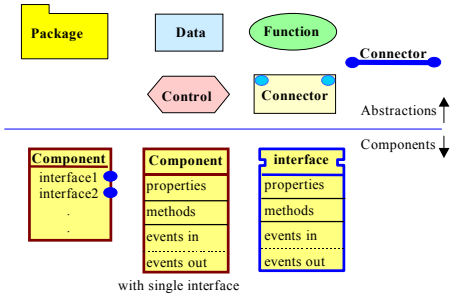
Modeling Emphasis for Different Approaches



A New Paradigm

- Build by Integration vs. code writing
- Issues
  - Locating the components
  - Integration
- Hard Engineering disciplines discovered earlier

Graphical Symbols in COSEML



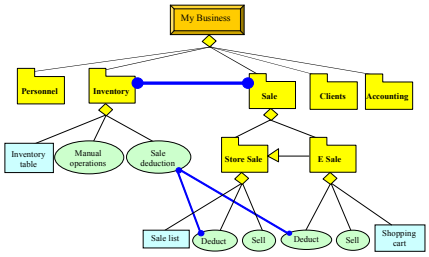
COSE Symbols and Their Meanings

	Package
	Function
	Data
	Control
	Connector
	Component

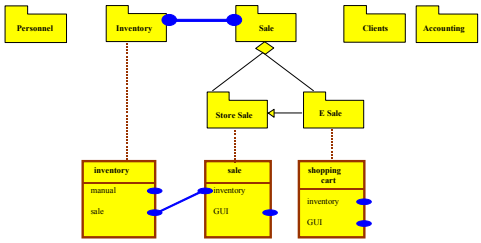
COSE Symbols and Their Meanings (contd.)

	Interface
	Represents
	Event link
	Method link
	UML …

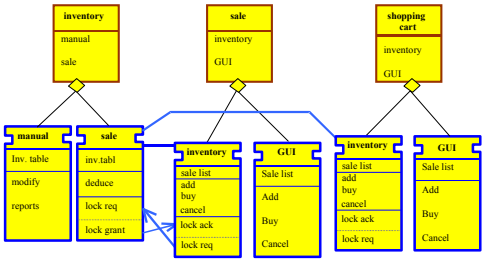
Abstract Modeling of a Small Business in COSE

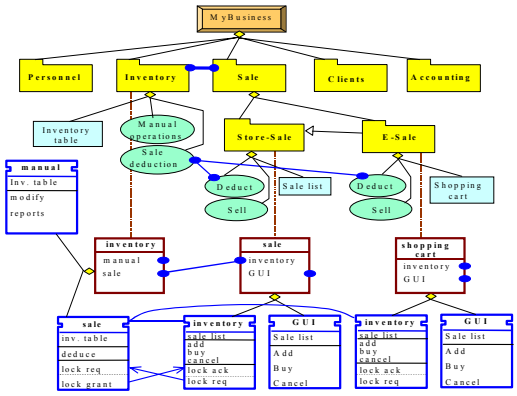


Corresponding Abstractions to Components in COSE



Detailed Modeling Through Components in COSE





A Case Study

- 20 projects
- UML and COSEML designs for each project
- Groups of 1-2 senior undergraduate students
- Students having a little O-O background and no UML or COSEML experience
- Two metrics forms

Project Titles

- Factory Automation System
- Inventory System for Small Business
- Component-Based Case Tool
- THBT Member Tracing
- Hospital Management Automation
- Data Warehouse Office Automation
- Travel Agency Automation
- Hospital Automation System
- Automile Stability Control System
- E-Commerce for Selling Mobile Phones

Project Titles (Contd.)

- Human Resources System
- Component-Based Case Tool
- Modeling of Dormitory Management System
- General E-Commerce
- Robot Pet Karabash
- Pharmacy Automation System
- Seat Reservation System for Laboratory Environments
- A Web-Based Application for a Multi-national Company
- Web-Based Teaching
- Computer Center Library Automation

Process Metrics

- Total development, modification, and correction efforts:
  - Effort: Person-hours spent for the development of a design model
  - Modification: Effort when any part of a design is revisited
  - Correction: Effort for revisions after the completion of the design

Process Metrics Results: Effort

Success Comparison based on Number of Projects

	Total	UML better	COSEML better	Equal	Not Applicable
Number of Projects	20	9	5	3	3
Percentage (%)	100	45	25	15	15

Success Comparison based on Statistics

	Number of Projects	Total Effort	Mean	Standard Deviation
UML	20	463	23.15	14.07
COSEML	17	365	21.47	12.66

Process Metrics Results: Modification

Success Comparison based on Number of Projects

	Total	UML better	COSEML better	Equal	Not Applicable
Number of Projects	20	6	6	4	4
Percentage (%)	100	30	30	20	20

Success Comparison based on Statistics

	Number of Projects	Total Effort	Mean	Standard Deviation
UML	18	80	4.44	3.76
COSEML	17	71	4.18	2.62

Process Metrics Results: Correction

Success Comparison based on Number of Projects

	Total	UML better	COSEML better	Equal	Not Applicable
Number of Projects	20	5	6	2	7
Percentage (%)	100	25	30	10	35

Success Comparison based on Statistics

	Number of Projects	Total Effort	Mean	Standard Deviation
UML	15	22.95	1.53	1.23
COSEML	15	31.50	1.85	1.78

Subjective Evaluation

Two subjective evaluations:

1. EASE: It was easy to model your problem using UML/COSEML.
2. Understandability: Your model is an understandable representation of the problem.

Process Metrics Results: Ease

Success Comparison based on Number of Projects

	Total	UML better	COSEML better	Equal
Number of Projects	20	6	5	9
Percentage (%)	100	30	25	45

Success Comparison based on Statistics

	Number of Projects	Total Grade
UML	20	77
COSEML	20	73

Process Metrics Results: Understandability

Success Comparison based on Number of Projects

	Total	UML better	COSEML better	Equal
Number of Projects	20	6	2	12
Percentage (%)	100	30	10	60

Success Comparison based on Statistics

	Number of Projects	Total Grade
UML	20	81
COSEML	20	75

Product Metrics

- Also called Model-Complexity Metrics
- Different for UML and COSEML
- Cannot be directly compared. Needs different strategies. Results left out of thesis.

Conclusions

- For the small projects involved, COSEML did not display a remarkable drawback,
- Further case studies are required, especially to be carried out by more experienced developers.

Similar Case Studies in the Field

- No similar case studies for component-software
- D. A., Boehm-Davis, L. S., Ross,"Program design methodologies and the software development process", International journal of Man-Machine Sudies, vol. 36, pp. 1-19, 1992.

  Compared:
  - Jackson program design
  - Object-oriented design
  - Functional decomposition.

  Used metrics to evaluate solution completeness, time to design and code, and model complexity.

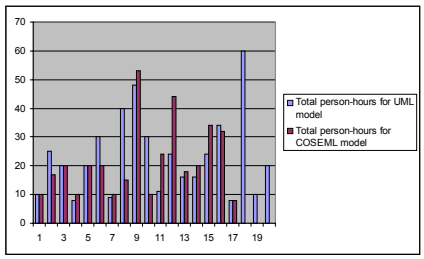
Product Metrics for Object-Oriented Models

- Number of Classes
- Number of Objects
- Number of Subsystems
- Number of Methods per Class
- Maximum Depth
- Maximum Width
- Number of Compositions
- Number of Inheritances

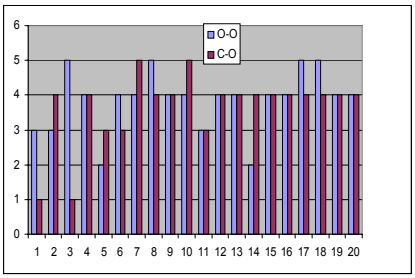
Product Metrics for Component-Oriented Models

- Number of Boxes
- Number of Event Links
- Number of Method Links
- Number of Methods per Component
- Number of Interfaces per Component
- Number of Methods per Interface
- Number of Subsystems
- Maximum Depth
- Maximum Width

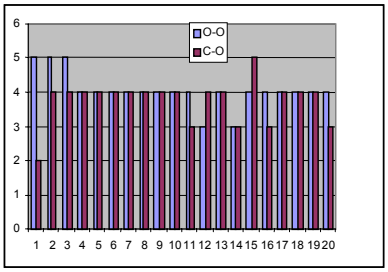
Effort Values for each Project



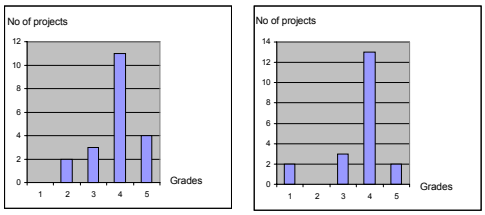
Ease Grades for each Project



Understandability Grades for each Project



EASE Grade Counts for UML and COSEML



Understandability Grade Counts for UML and COSEML