

Situational Awareness: Discovering the Dependencies



Software Dependency Analysis using DSM and UML Models

Sung-Hee Do dosh@axiod.com – (617) 746-9222 ext 202,
Julie Carignan carignan@axiod.com – (617) 746-9222 ext 204

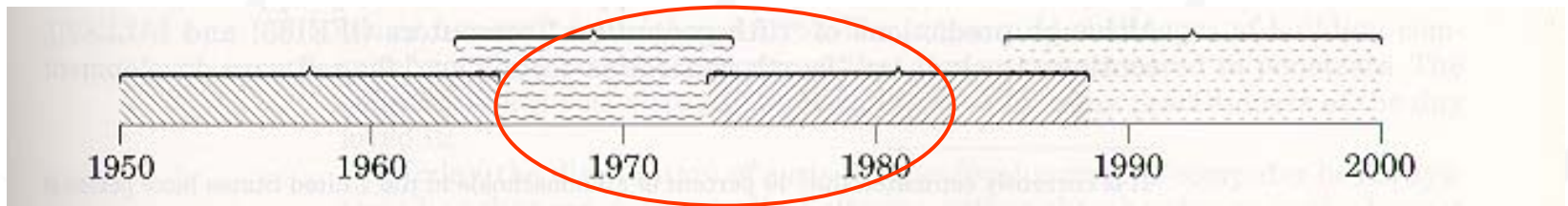
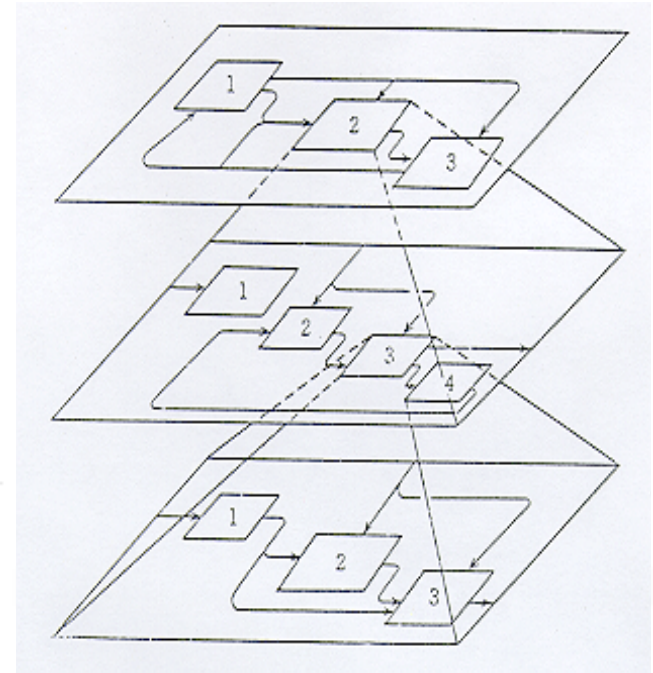
Outline

- Introduction
- Unified Modeling Language (UML)
- Creating dependency matrix using Reverse Engineering
- DSM analysis for software dependency
- DSM software – Acclaro Scheduler
- Case Study

Software Design:

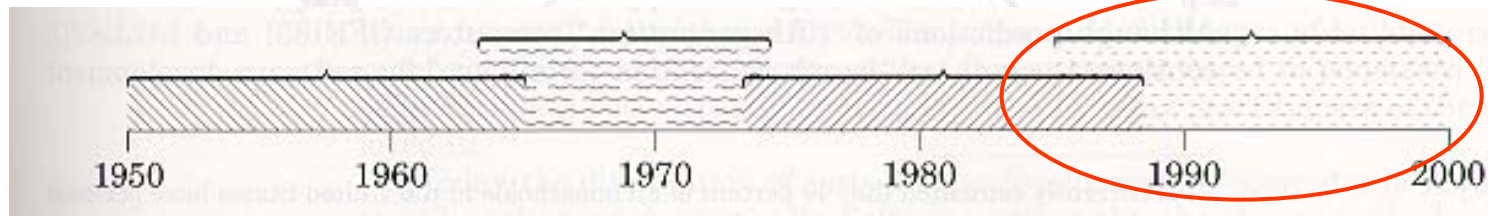
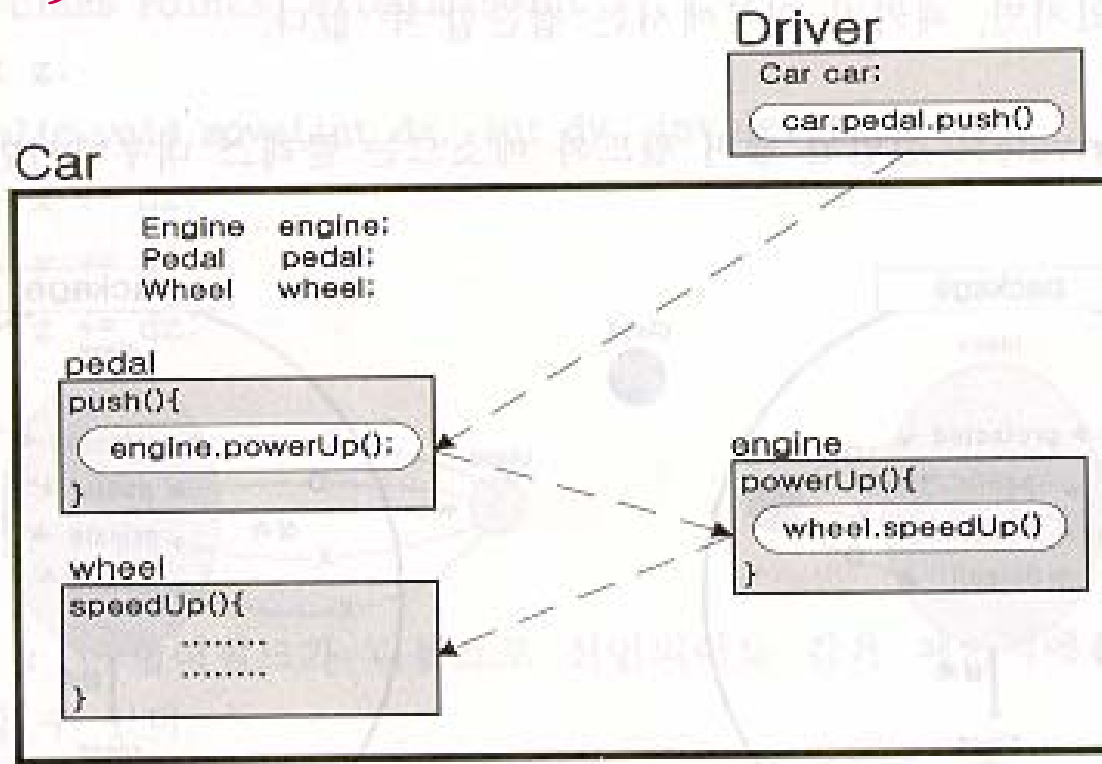
Overview - Structured programming method

- Modular programming
 - Function abstraction
 - Requires data structure
- Top-Down programming
- Less reusability for the developed module



Software Design:

Overview - Object-Oriented programming (1/2)



Software Design:

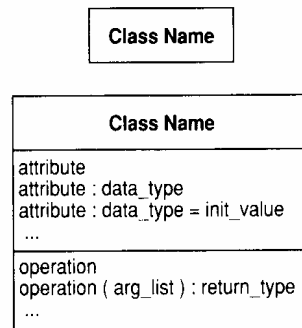
Overview - Object-Oriented programming (2/2)

- Modular programming
 - Data abstraction
 - Data(attribute) is included in each module
- Bottom-Up programming (component base)
- Each object is reusable
- Relationship (e.g. association, whole-part, generalization-specialization) between each objects is much more important

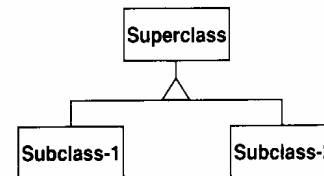
Unified Modeling Language

➤ Rumbaugh's Object Model Notation

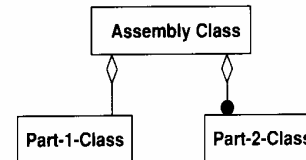
Class:



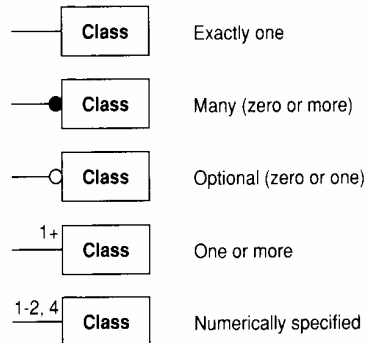
Generalization (Inheritance):



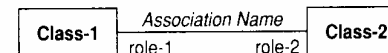
Aggregation:



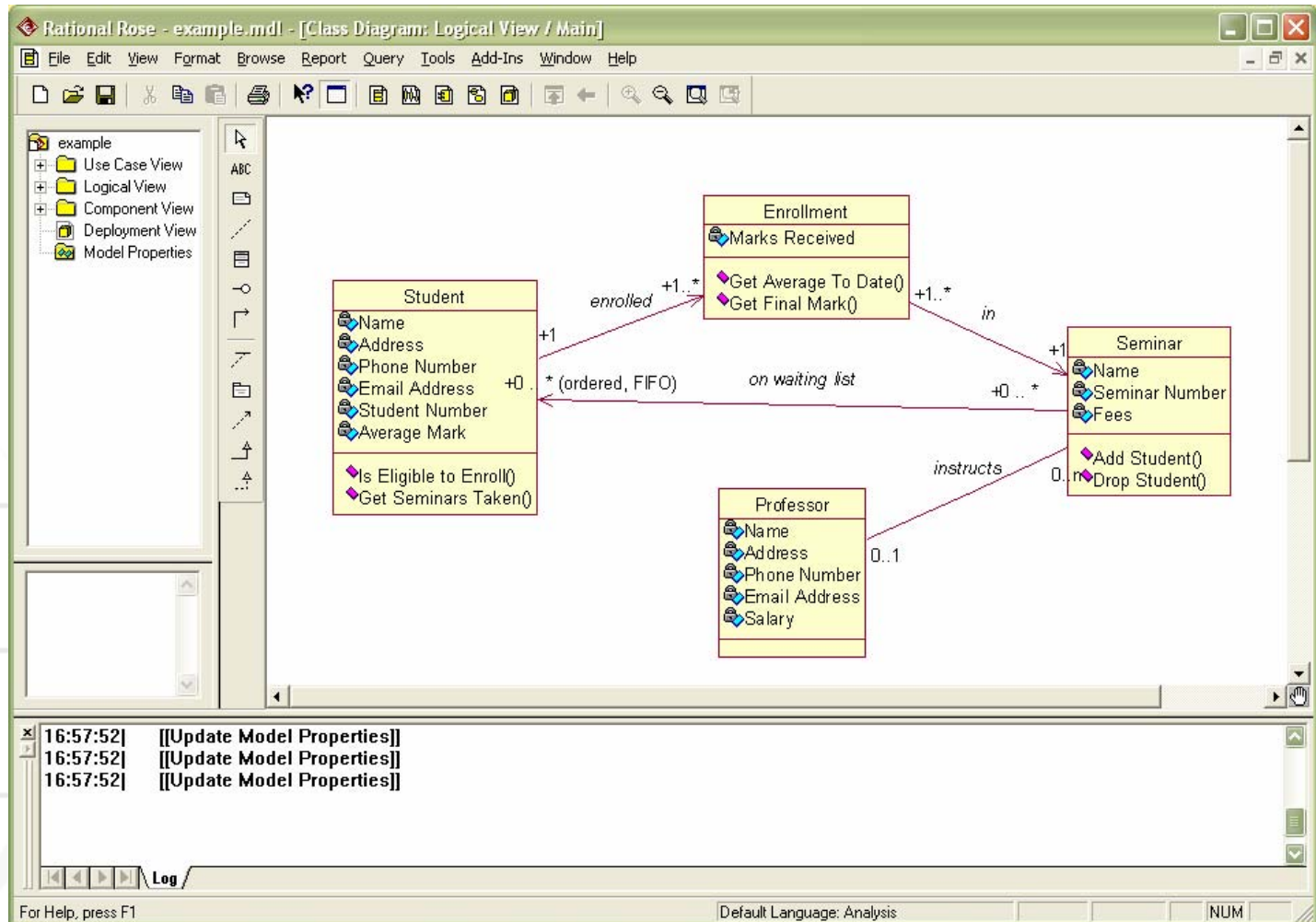
Multiplicity of Associations:



Association:



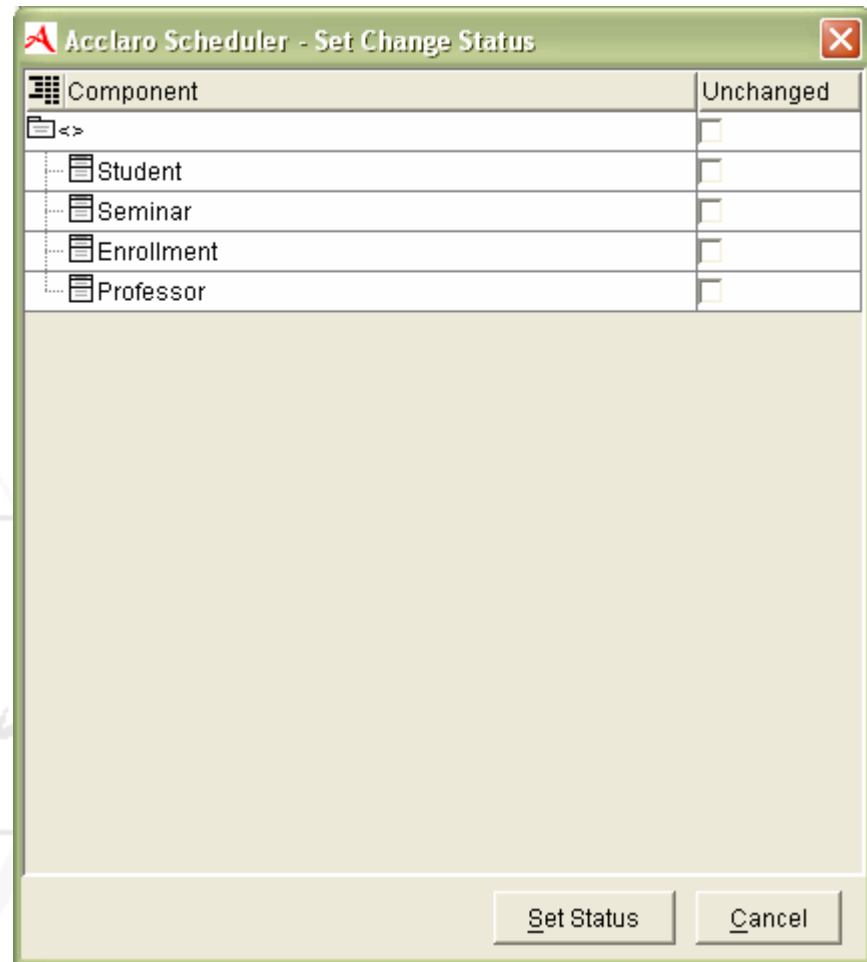
Class Diagram



Creating DSM

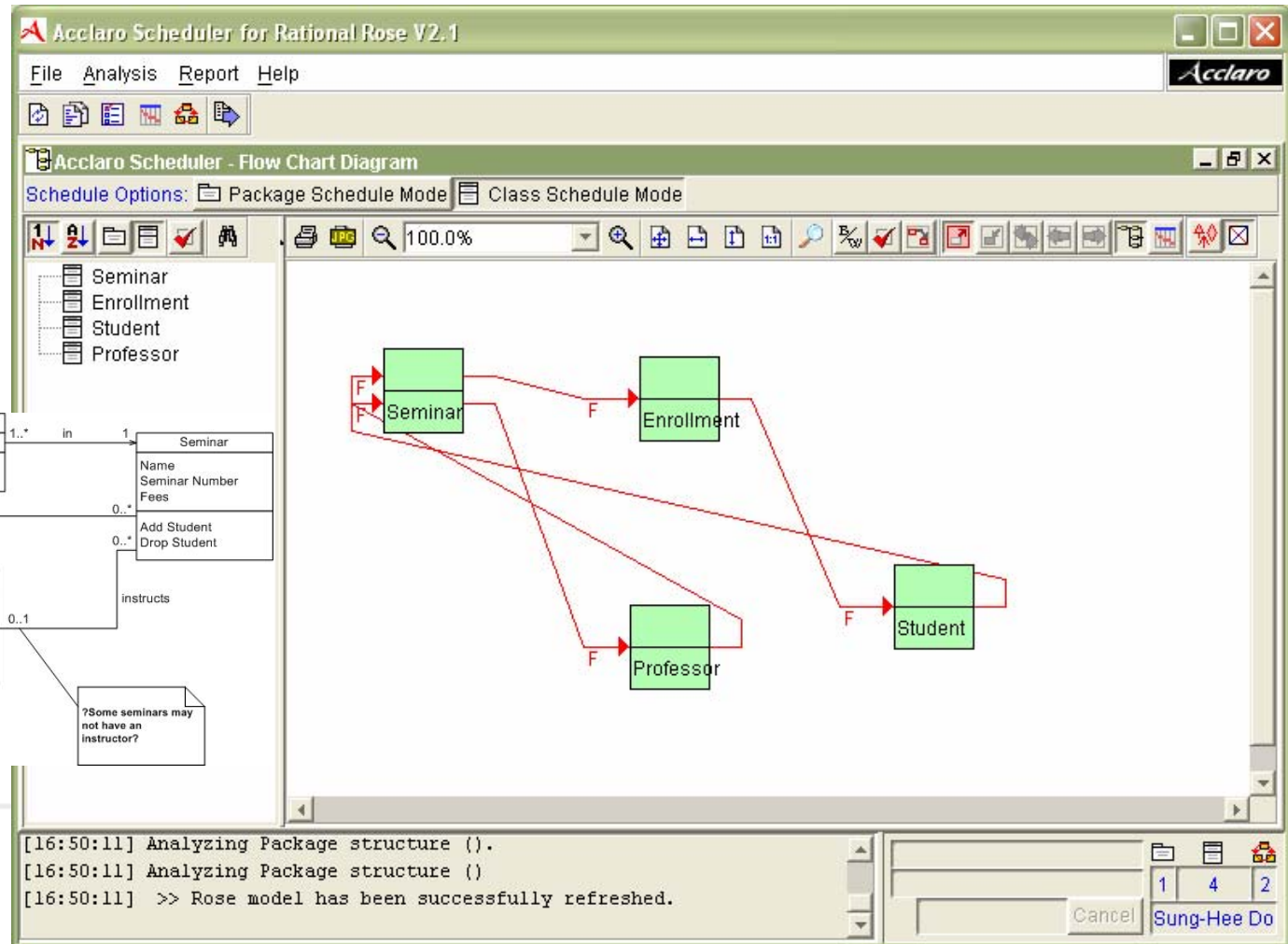
Step 1 – Identifying Classes (2/2)

- Student
- Enrollment
- Seminar
- Professor

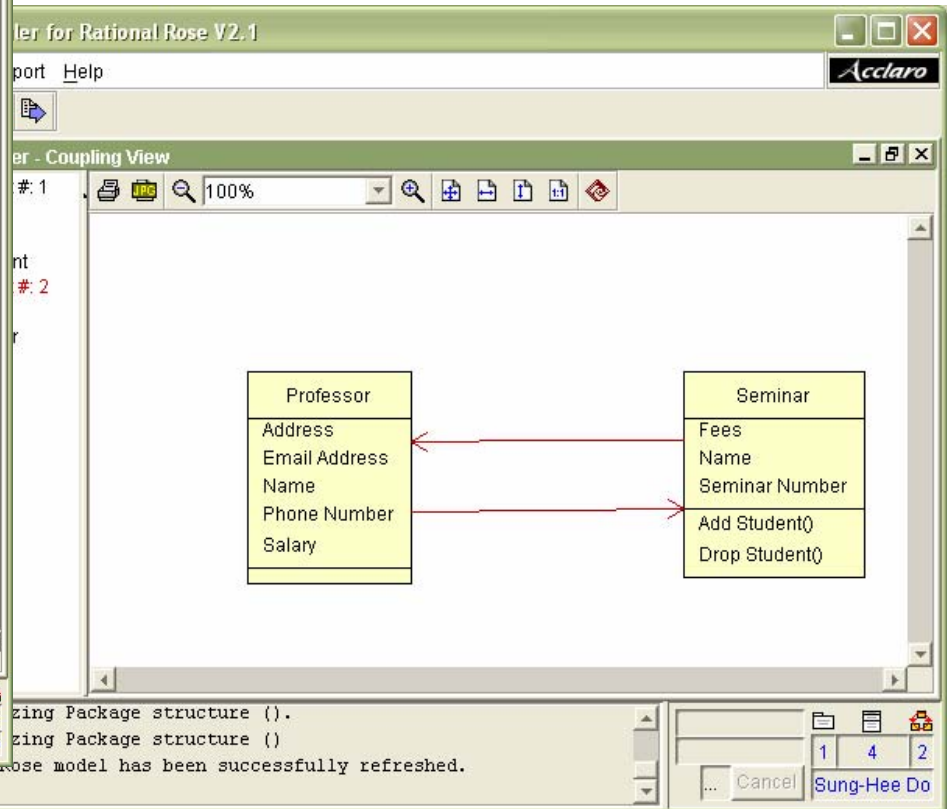
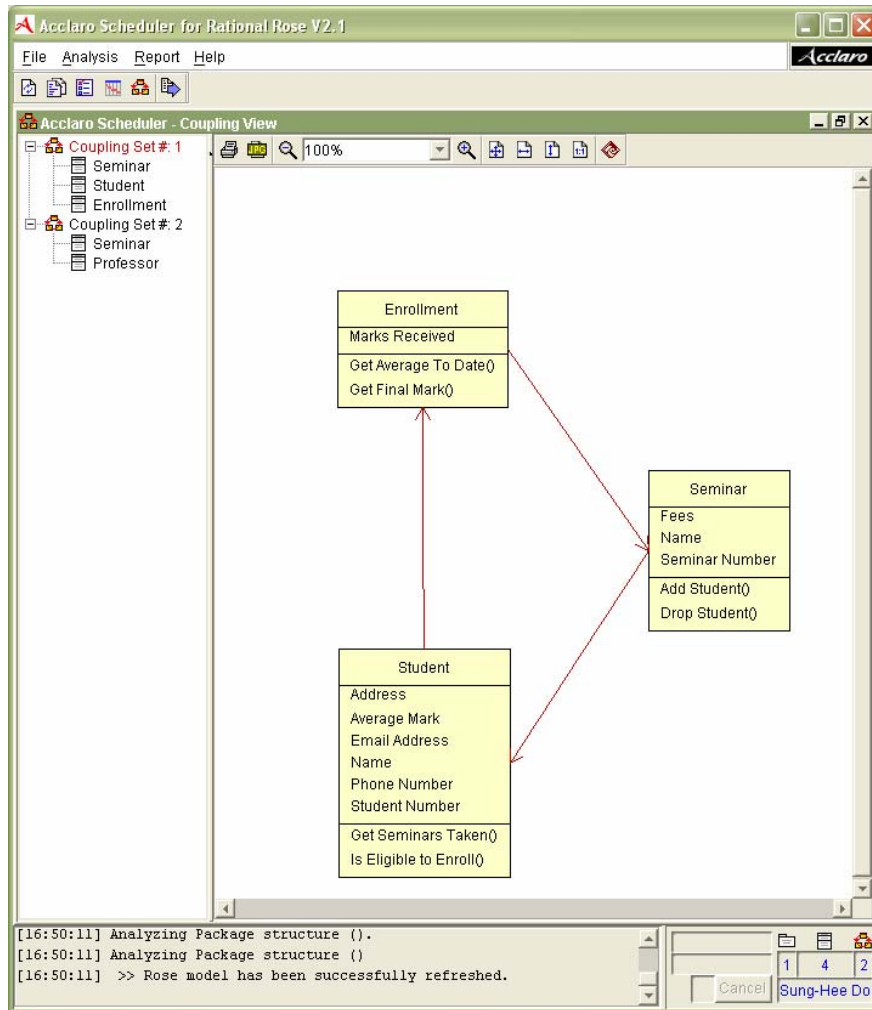


Creating DSM

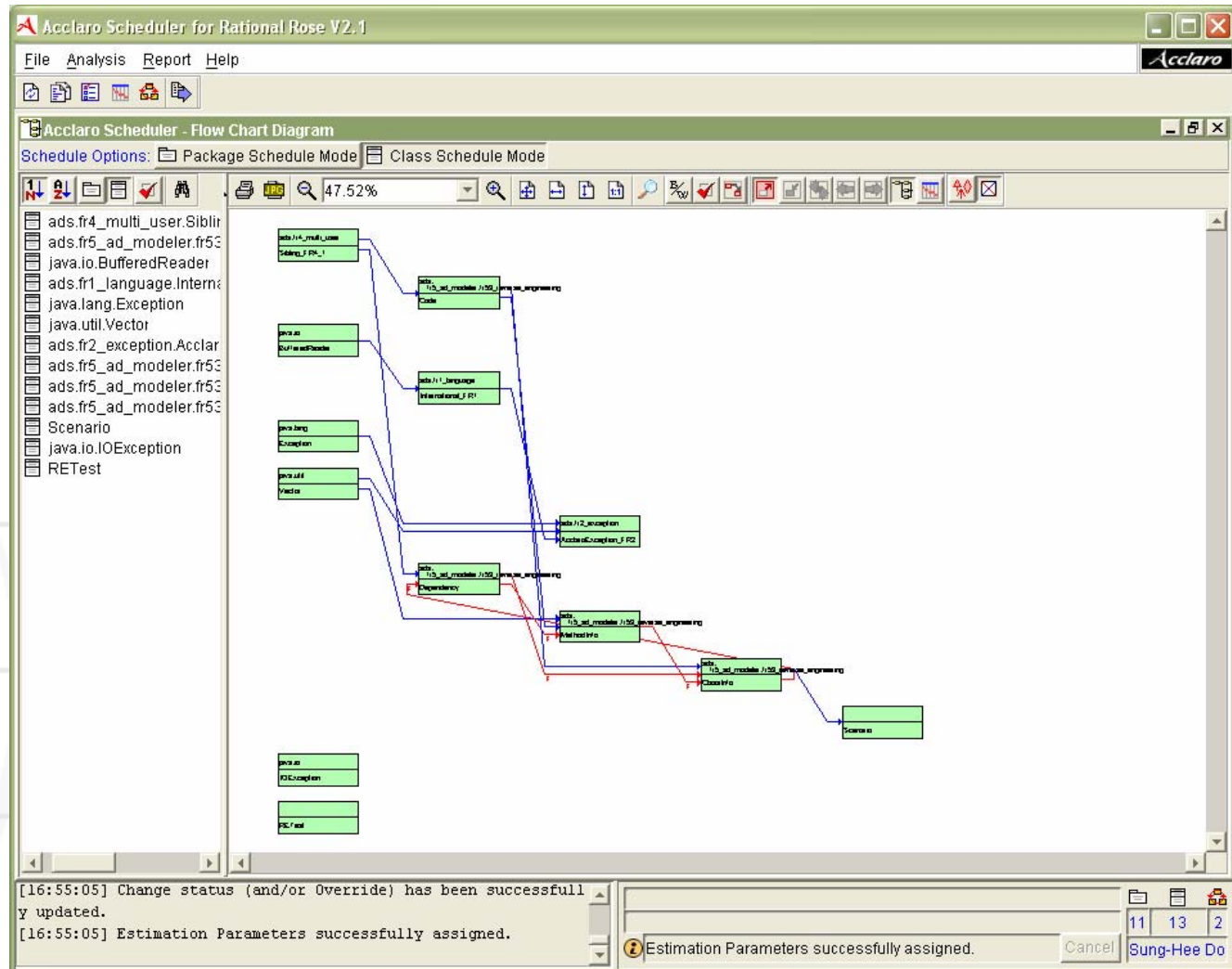
Step 2 – Identifying Relationships (2/2)



DSM analysis: Clustering



DSM analysis: Sequencing/Partitioning



DSM analysis:

Estimation for project schedule

Acclaro Scheduler - Estimation Parameters

Component	Base	Remaini...	Stereoty...	Attrs	Ops	Rels	Unchan...
<re.mdl>	39.12	39.12		13	103	14	<input type="checkbox"/>
ads(7)	35.28	35.28		13	90	13	<input type="checkbox"/>
fr4_multi_user	3.41	3.41		1	17	0	<input type="checkbox"/>
Sibling_FR4_1	3.41	3.41		1	17	0	<input type="checkbox"/>
fr5_ad_modeler(4)	26.98	26.98		6	56	9	<input type="checkbox"/>
fr53_reverse_engineering(4)	26.98	26.98		6	56	9	<input type="checkbox"/>
Code	4.85	4.85		4	20	1	<input type="checkbox"/>
Dependency	6.78	6.78		2	12	2	<input type="checkbox"/>
MethodInfo	7.68	7.68		0	12	3	<input type="checkbox"/>
ClassInfo	7.68	7.68		0	12	3	<input type="checkbox"/>
fr1_language	1.94	1.94		2	8	1	<input type="checkbox"/>
International_FR1	1.94	1.94		2	8	1	<input type="checkbox"/>
fr2_exception	2.94	2.94		4	9	3	<input type="checkbox"/>
AcclaroException_FR2	2.94	2.94		4	9	3	<input type="checkbox"/>
java(4)	0.8	0.8		0	0	0	<input type="checkbox"/>
int(2)	0.4	0.4		0	0	0	<input type="checkbox"/>

Stereotype Specific: **Global**

Global Stereotype for Scheduling (Preference)

Days per Attribute:

Days per Method:

Relationship Weight Factor:

Coupling Weight Factor:

Revalidation Factor:

Minimum Duration:

Description

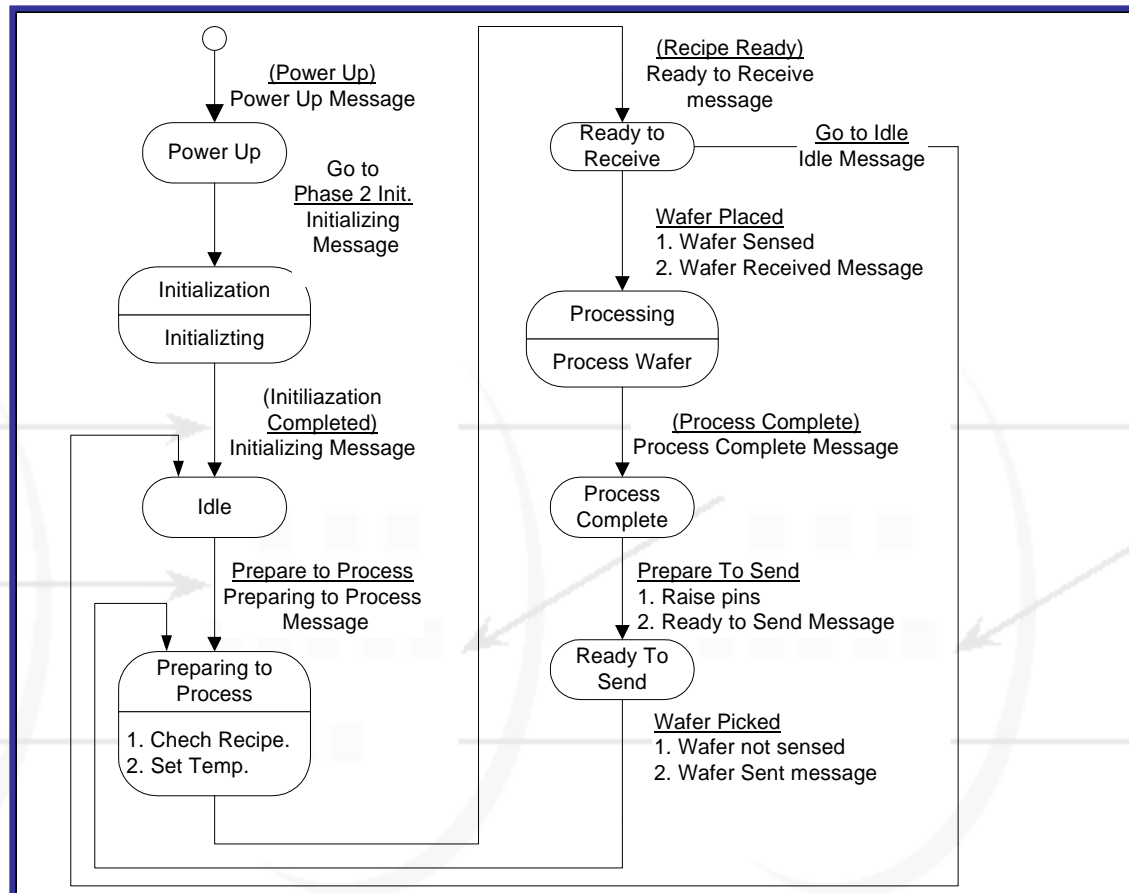
This page allows you to fine tune the general estimation parameters. You can adjust the parameters and use the "Simulate" button below to recalculate the estimated duration. You can also override each value on a class by class basis by manually entering a duration in the corresponding cell in the table above, or by using the "Stereotype" tab to override the duration for all classes of a given stereotype. Use the "Apply" or "OK" buttons below to save your changes.

Simulate Apply OK Cancel

Case Study:

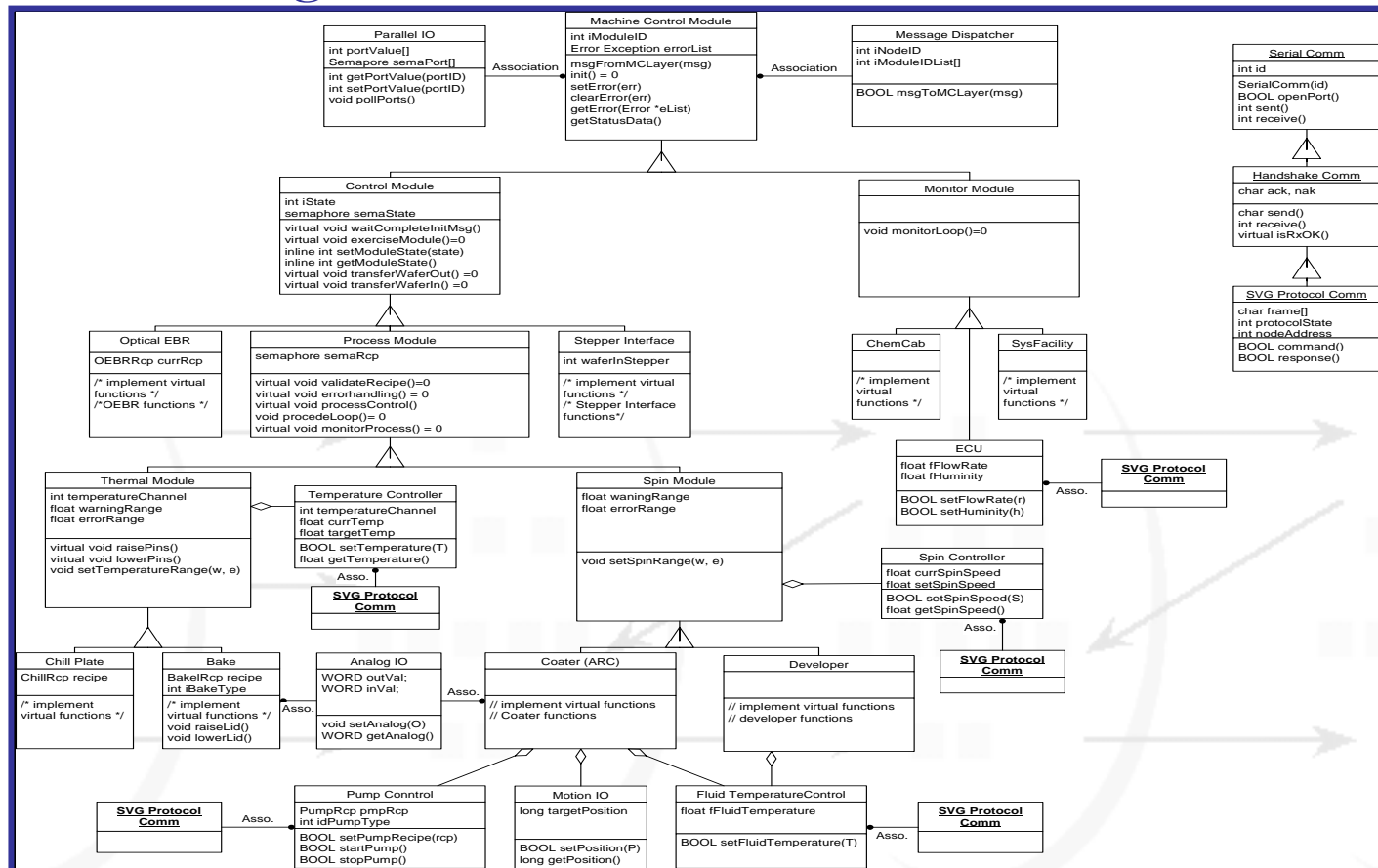
Chill plate control module (SVG Track)

➤ State Diagram



Case Study: Chill plate control module (SVG Track)

➤ Class Diagram



Case Study:

Chill plate control module (SVG Track)

- Translation process for reversing from OO to DSM
- Scan the UML model to construct the diagonal term of the matrix
 - Classes, Packages
 - Scan the source code again to construct the off-diagonal term of design matrix
 - Find the operation call inside the method
 - Mark the operation call into the matrix

↗ Design Structure Matrix

AXIOMATIC DESIGN™
SOLUTIONS, INC.

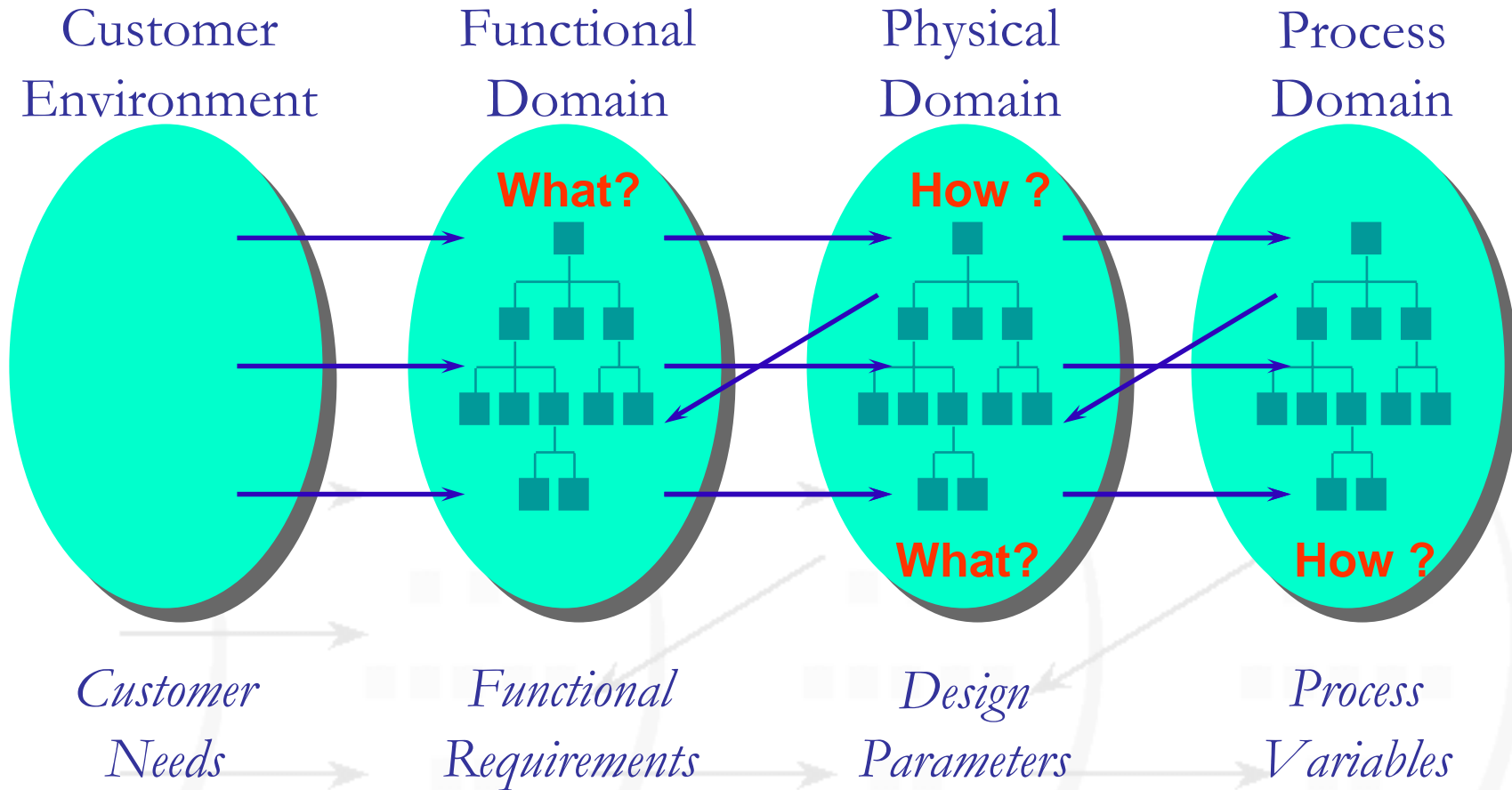
Extending DSM to use cases

	U1	U2	U3	U4	U5	U6	U7
U1	*	X			X	X	
U2		*		X			X
U3		X	*	X			X
U4		X	X	*	X		X
U5				X	*	X	
U6	X				X	*	
U7		X	X	X			*

Requirement handling using clustering

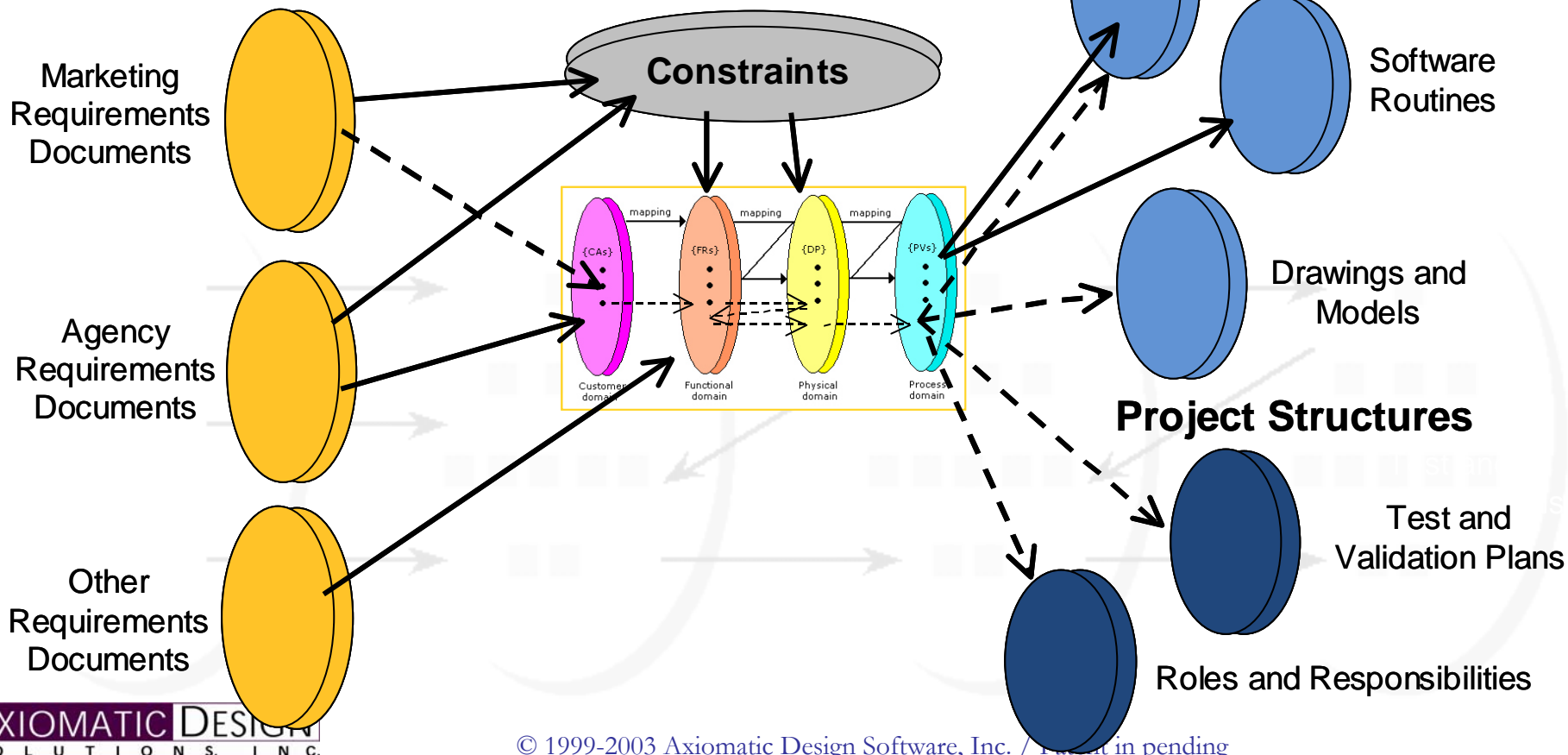
	U1	U6	U5	U4	U2	U3	U7
U1	*	X	X	X			
U6	A	*	X				
U5		X	*	X			
U4			X	C*	X	X	X
U2				X	*	D	X
U3				X	X	*	X
U7				X	X	X	*

Axiomatic design: Mapping, hierarchies, and zigzagging



End to End Requirements Traceability

External Requirements



Discussion

- An addin software for Rational Rose has been developed.
- DSM analysis for Software system gives benefits for work load grouping using clustering technique, scheduling and estimation using partitioning.
- The axiomatic process along with DSM concept defines a repeatable, yet flexible solution to many pitfalls in the area of software product lifecycle management.