



A UML-based Generic Set of Modeling Rules

Michel Lizotte

Defence Research Establishment Valcartier (DREV)

2459 Blvd. Pie XI (North), Val-Bélair, Québec, Canada G3J 1X5

e-mail: Michel.Lizotte@drev.dnd.ca



National
Defence

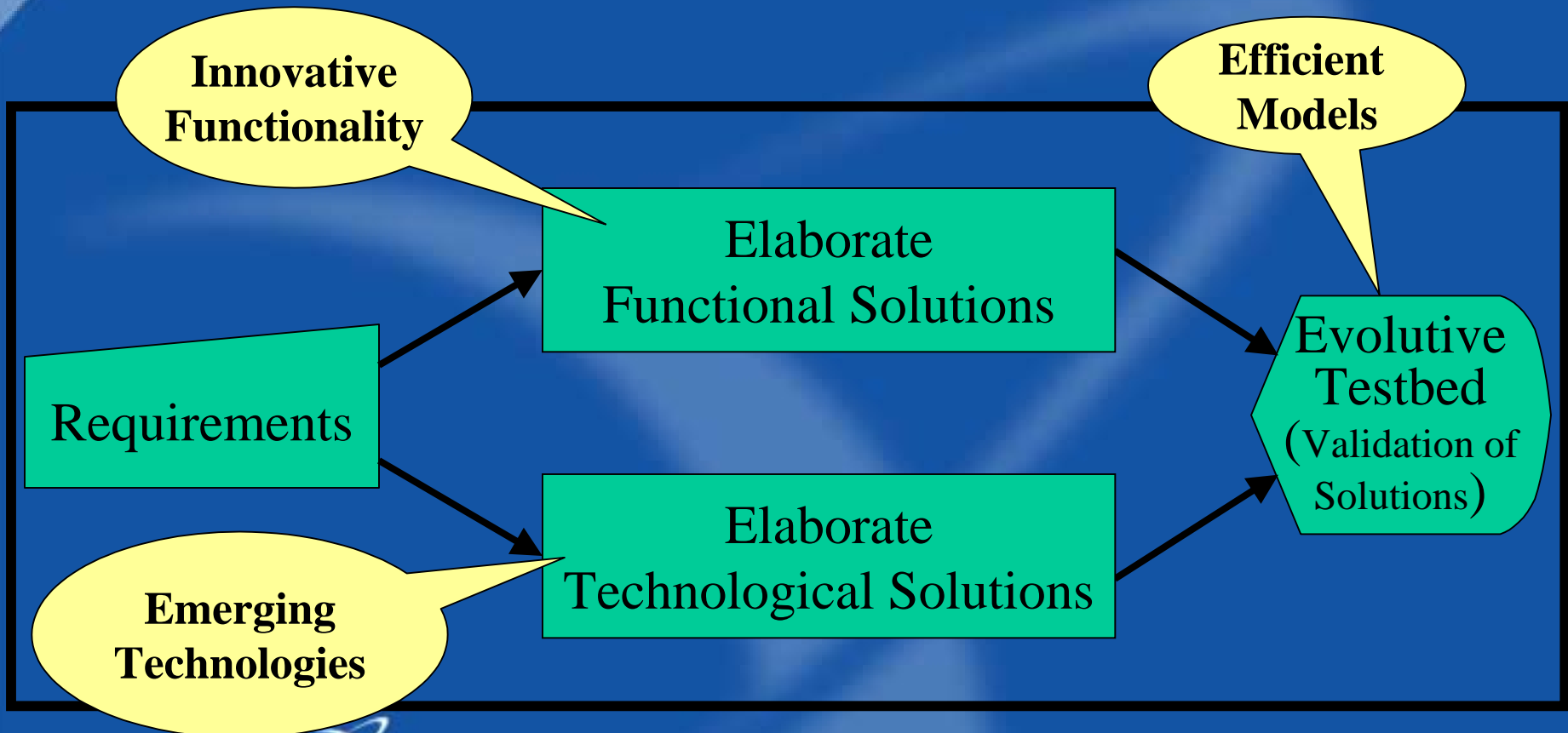
Défense
nationale

Canada

Presentation Plan

- Context
- Overview
- Top-level rules
- Diagram rules
- Summary

From: Requirements To: Validated Software Models



1

- I want functionality and usability.
- I am not interested in your blueprints.
- That is not what I was expecting.

3

- The documentation I paid for is never used!
- I do not like unexpected costs.
- How could I keep this knowledge?



The Manager



The User

4

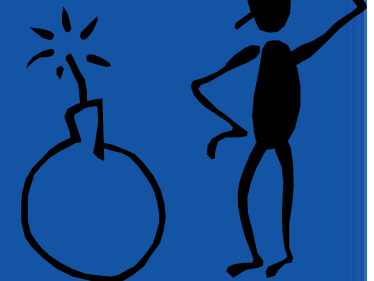
The Mission:
Effective Modeling
and Documentation

2

- This documentation is tedious.
- I'm leaving tomorrow.
- I have an opportunity (\$\$\$) to try new technologies elsewhere.

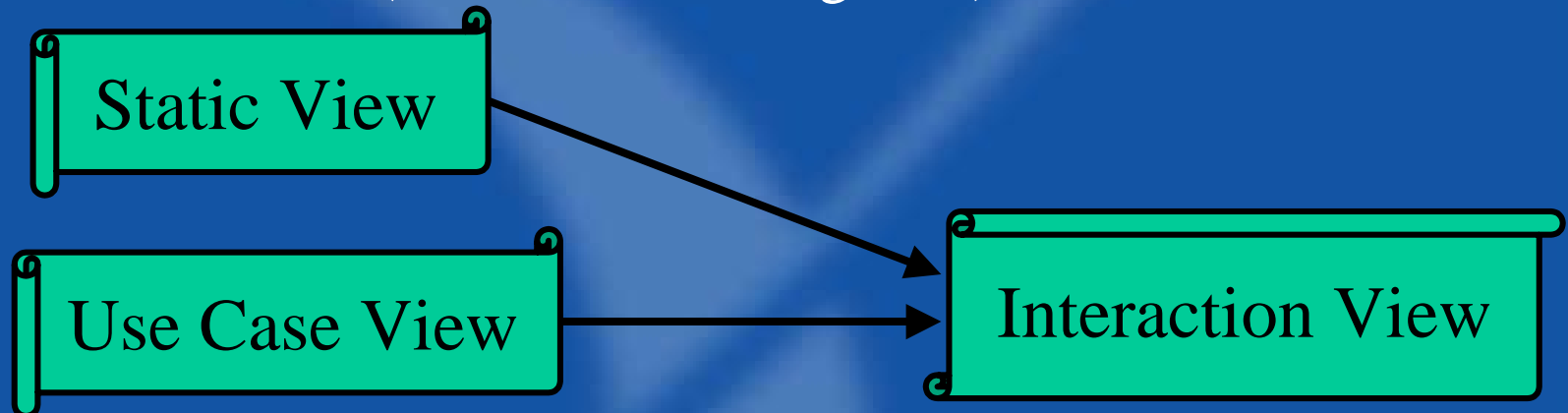


The Developer



Master principles of the approach

- Popping components using two distinct perspectives
 - Object (oo developers looking at data)
 - Behaviour (oo developers looking at functionality)
- Delaying identification of class operations until design of service units (collaboration diagrams).



UML models

Object
perspective

- Object Architecture Model
 - stable structure of conceptual level classes and relationships (well defined and stable)
- Object Design Model
 - implementation level classes with stable attributes and operations
- Database Design Model
 - persistence implementation structure addressing integrity and performance issues

UML models (cont'd)

Behaviour
perspective

- Behaviour Architecture Models
 - stable partitioning of functionality offered by the system
- Behaviour Design Models
 - stable implementation logic for services

Object levels

UML extensions
Package stereotypes

- 1st level => System Object
- 2nd level => Object Division
- 3rd level => Object Section

UML
Class

- 4th level => Class

Behaviour levels

UML extensions
Use Case stereotypes

- 1st level => System Service
- 2nd level => Service Division
- 3rd level => Service Section
- 4th level => Service Group
- 5th level => Service Unit

UML extensions
Operation stereotypes

- 6th level => Master Operation
- 7th level => Subordinate Operation

b1 System Service

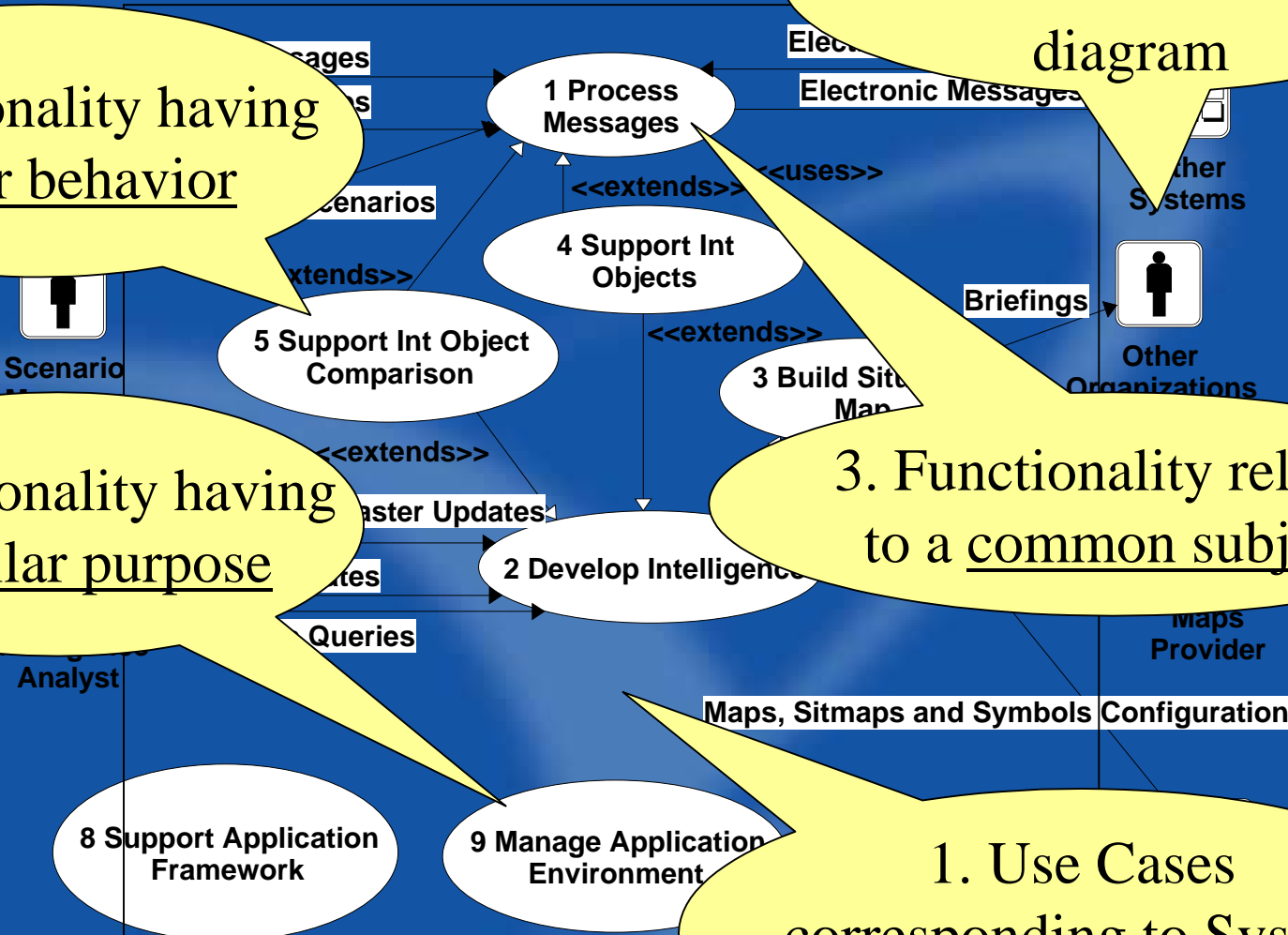
4. Functionality having similar behavior

2. Same Actors identified in the context diagram

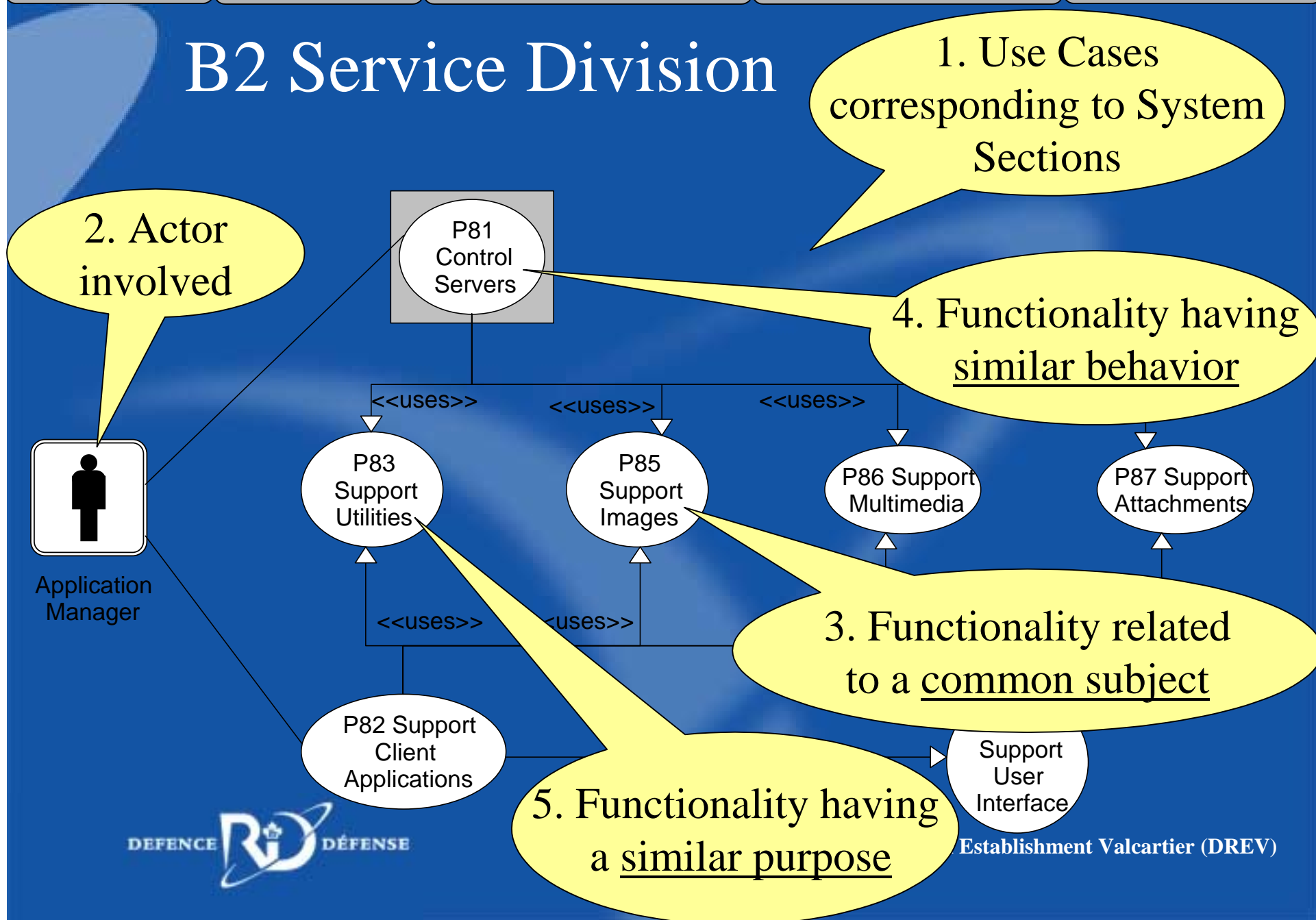
5. Functionality having a similar purpose

3. Functionality related to a common subject

1. Use Cases corresponding to System Divisions



B2 Service Division



1. Use Cases corresponding to System Sections

2. Actor involved

4. Functionality having similar behavior

3. Functionality related to a common subject

5. Functionality having a similar purpose

b3 Service Section



Intelligence Analyst

P311 Manipulate Sitmap View List

1. Use Cases corresponding to Service Groups

<<extends>>

P313 Filter Sitmap View

<<uses>>

P312 Handle Sitmap View

<<extends>>

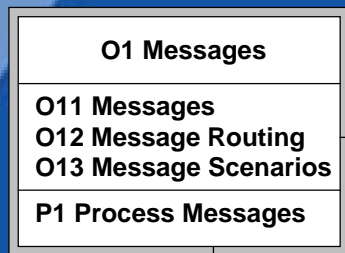
P314 Preview Symbols

2. A functionality that need to be separately documented and controlled

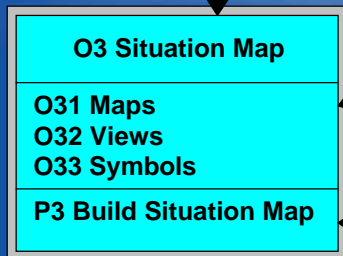
From Service Group to Service Units

- a user interface (e.g. Handle Unit)
- a complex functionality (e.g. Compute Combat Effectiveness)
- an independent module (e.g. Read Message Text Format)
- an intersystem interface (e.g. Control Server Public Interface)
- an intrasystem interface (e.g. Control Server Limited Interface)
- a client part (e.g. Share Overlays)
- a server part (e.g. Control Overlay Sharing)

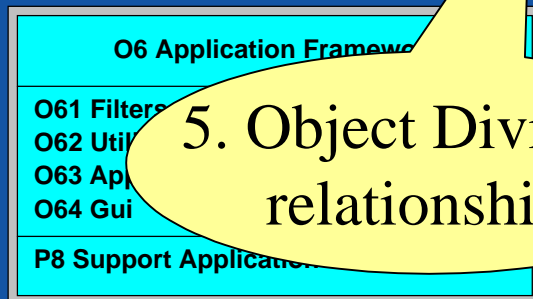
o1 System Object Diagram



are shown on



are shown on

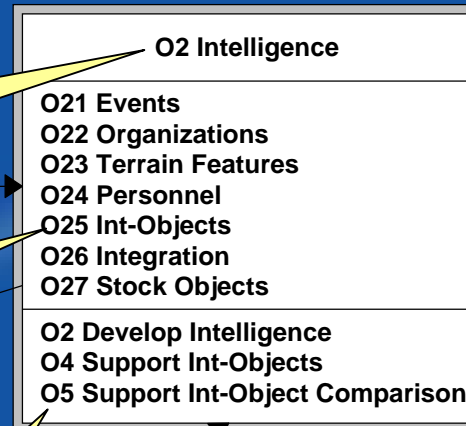


5. Object Divisions relationships

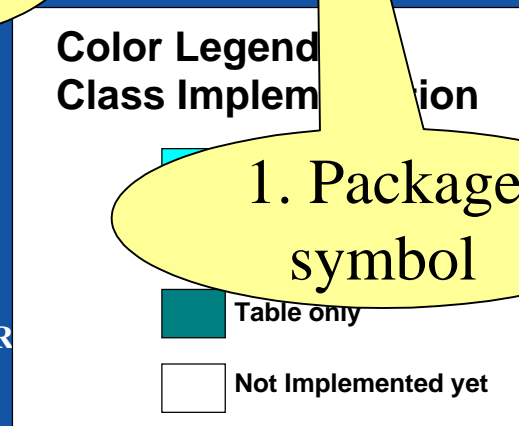
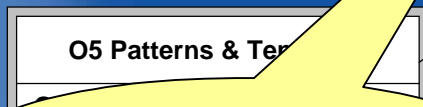
2. Object Division name

3. list of Object Sections

4. list of Service Divisions



generates



1. Package symbol

o3A Object Section Architecture

