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
Presentation Plan

• Context
• Overview
• Top-level rules
• Diagram rules
• Summary
From: Requirements
To: Validated Software Models

- Innovative Functionality
- Efficient Models
- Evolutive Testbed (Validation of Solutions)
- Elaborate Functional Solutions
- Elaborate Technological Solutions
- Requirements
- Emerging Technologies
Context

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

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

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

4. The Mission:
   Effective Modeling and Documentation

Overview
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 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  \( \Rightarrow \) System Object
- 2nd level  \( \Rightarrow \) Object Division
- 3rd level  \( \Rightarrow \) Object Section
- 4th level  \( \Rightarrow \) Class
Behaviour levels

- 1st level ⇒ System Service
- 2nd level ⇒ Service Division
- 3rd level ⇒ Service Section
- 4th level ⇒ Service Group
- 5th level ⇒ Service Unit
- 6th level ⇒ Master Operation
- 7th level ⇒ Subordinate Operation
1. Use Cases corresponding to System Divisions

2. Same Actors identified in the context diagram

3. Functionality related to a common subject

4. Functionality having similar behavior

5. Functionality having a similar purpose
B2 Service Division

1. Use Cases corresponding to System Sections
2. Actor involved
3. Functionality related to a common subject
4. Functionality having similar behavior
5. Functionality having a similar purpose
**b3 Service Section**

1. Use Cases corresponding to Service Groups

- **P311 Manipulate Sitemap View List**
- **P312 Handle Sitemap View**
- **P313 Filter Sitemap View**
- **P314 Preview Symbols**

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

**Intelligence Analyst**

Defence Research Establishment Valcartier (DREV)
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)
o3A Object Section Architecture

1. Class under the section

2. Class under another section

3. Service Groups (not operations)

Defence Research Establishment Valcartier (DREV)