10th ICCRTS

International Command and Control Research and Technology Symposium

On the Building of a UML Profile for the Description of Army Architectures in the Context of Complex Systems

Presented by Éric Dorion

Mario Couture and Antoine Duval
DRRC-Valcartier
(June, 2005)

Summary

Context and problematic

Present and future needs

Elements of solution

Conclusion

Context and Problematic (1/4)

Context: Canadian military acquisition:

- Taking into account revolution in military affairs (3-block war, asymmetric threats, …)
- Considering increased capability of electronic communications, software, hardware, people, …
- Considering new associated complexities

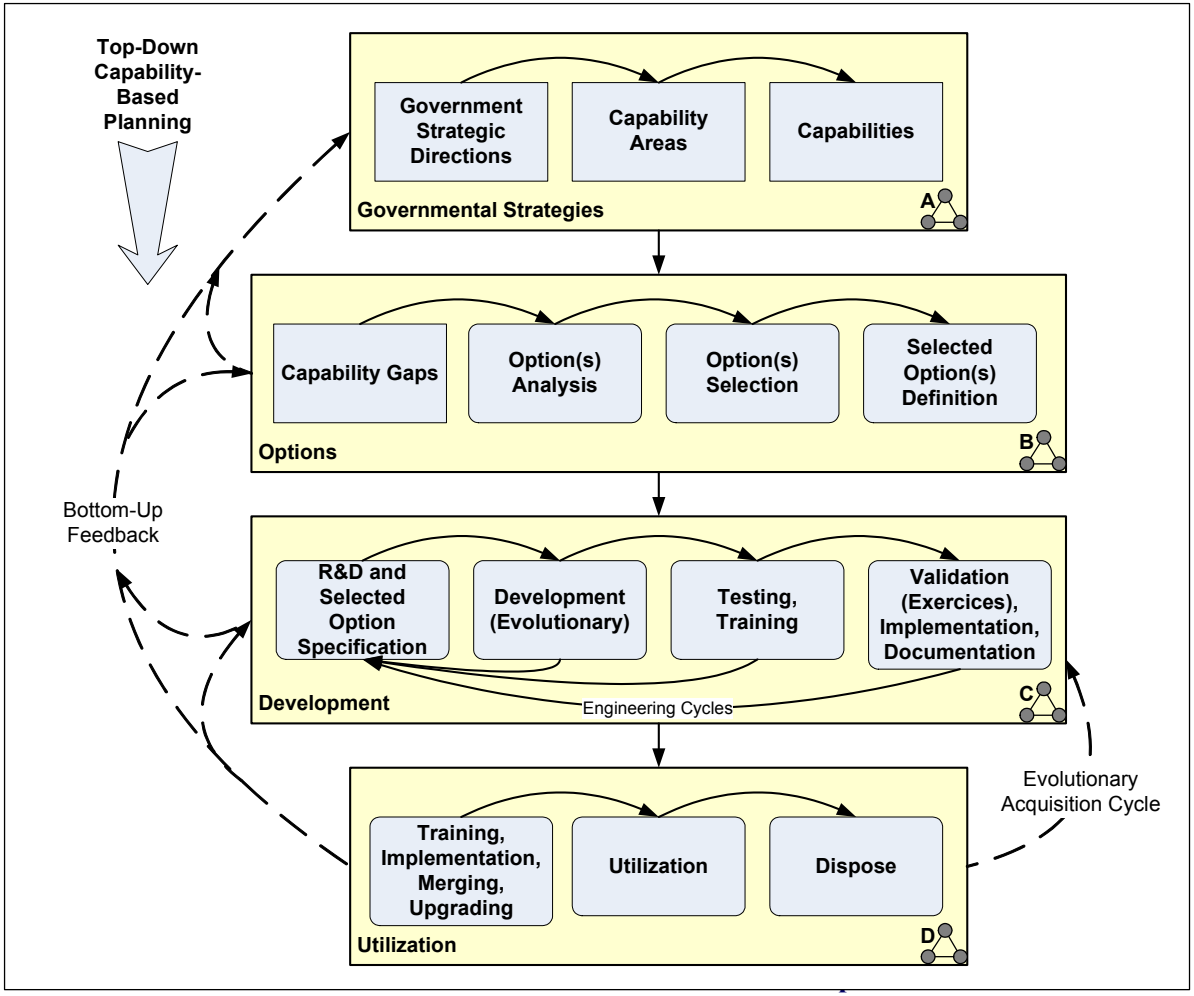
➔**Capability-Based Planning** to replace threat-based planning

Context and Problematic (2/4)

Capability-Based

Middle-out approach:

Both **top-down** and **bottom-up** are needed



Context and Problematic (3/4)

Thinking in terms of "Capability" involves:

- Being able to **use**, **re-use,** and **merge** autonomous systems and make them **collaborate** to produce capabilities

  →These form a complex **System of Systems (SoS)**

- Conceive, develop, and update all systems accordingly
- Better understand involved **complexities** and better control SoS behaviors during operations
- Consider the possibility of evolutionary approach for partial deliveries of capability

Context and Problematic (4/4)

In this simple example:

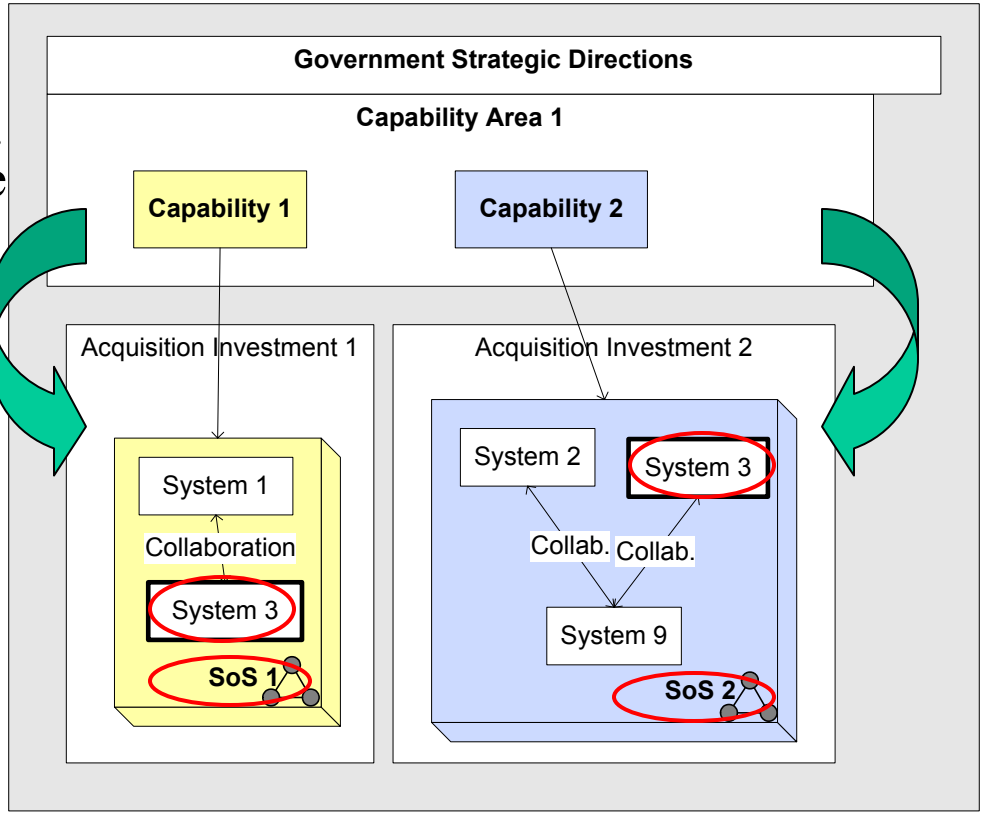
System 3 will be involved in two different SoS (1 and 2). It may contribute to achieve two different capabilities

➔**Three kinds of problems can be raised:**

➔Non-synchronization of efforts between acquisition projects

➔The non-availability of information from other acquisition projects

➔The misunderstanding of this shared information



Present and Future Needs (1/2)

To address this problematic:

➔**Model and link** any relevant domain element and information that may have direct and indirect influences on the whole (at enterprise level)

➔**Capture** (and dynamically keep updated) this set of linked models within a database and CASE tools (the architecture description)

➔**Integrate** and link other models/meta-models/DB in this architecture description

➔**Achieve complex searches** among this database

➔**Produce holistic diagrams** (linked models of different nature) that show all aspects that need to be viewed

Present and future needs (2/2)

The main needs are:

- Use a **holistic approach** (instead of reductionism)
- Revisit traditional linear and stovepipe System Engineering disciplines
- Consider new theories like System Thinking, complexity theory, etc
- **➔Revisit traditional way of describing architectures**

  ➔➔Modeling languages should:

  - **Be easy to use and understand by any stakeholder**
  - **Be flexible (allow the modeling and linking of any concept)**
  - **Support new ways of doing engineering**
  - **Be fully supported by CASE tools**

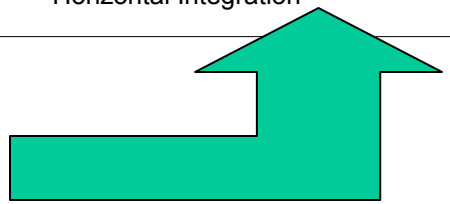
Elements of Solution (1/8)

The used "architectural levels" for describing military architectures:

Architectural Levels	Involved Stakeholders	Objects of Concern	Engineering Domain	Architecting Domain	Managmnt Domain	Other Domains
Operational Level	End-Users	Business Objects & Processes	Business-related models	Business-related models	Business-related models	Business-related models
System Level	Engineers/ Architects	System Design	Design-related models	Design-related models	Design-related models	Design-related models
Implementation Level	Developpers	Code files, Executables, Libraries	Devel.-related models	Devel.-related models	Devel.-related models	Devel.-related models

Vertical Integration

Horizontal Integration



Holism: starts with the **2-dimensional integration** (through UML and CASE tools) of all relevant domain elements and information

Elements of Solution (2/8)

The solution adopted:

➔**UML** (and its **extension mechanism)** was chosen as modeling language to support the description of military architectures

- UML is relatively known and accepted, it evolves with needs (SYSML), it is well supported by CASE tools, …

A **UML profile** had to be conceived and developed to better model and link relevant domain elements that are of different nature (people, processes, technologies, and materiel)

- This work is related to the building of the **Military Architecture UML Profile** (the **MAU-Profile**)

Elements of Solution (3/8)

We use a dynamic definition of "system" and "SoS":

System: a system is made of **people** (person, group, association, organization, etc) that use **processes** (doctrines, standards, methods, etc), **technologies** (software, frameworks, standards, etc), and **materiel** (physical tools, vehicles, etc) to transform **inputs** into **outputs** within specific **contexts** and under specific **rules**

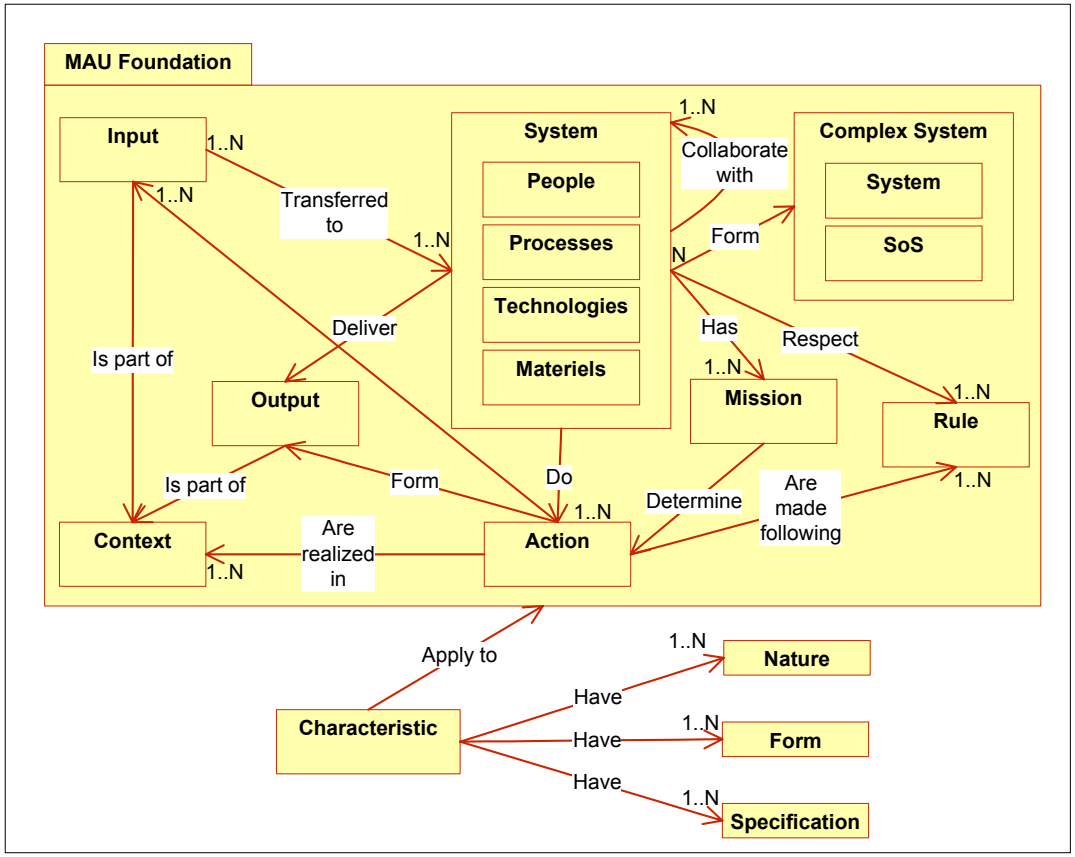
SoS: a SoS is an assemblage of normally **autonomous and independent systems** that **collaborate** with each other in order to get the ability to achieve a **mission-oriented set of actions** that allow the achievement of a **global mission**. This mission is understood and shared by all participating systems

Elements of Solution (4/8)

The MAU-Profile is made of many **stereotype names** that add UML model elements appropriate military semantic

The **foundation** of the MAU-Profile is mainly based on our dynamic definition of "System"

It defines **8 main entries** specifying how stereotype names can be classified (and retrieved)

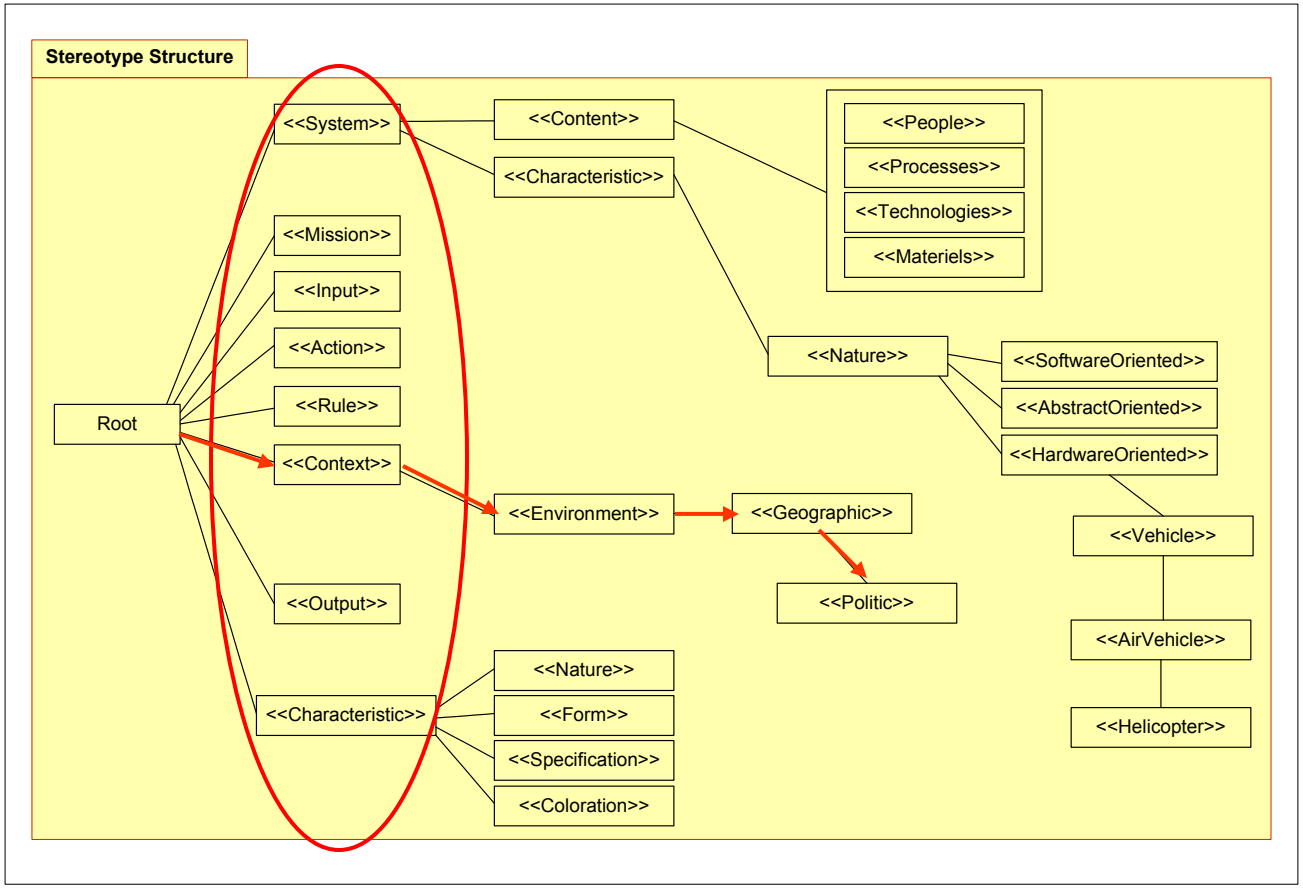


Elements of Solution (5/8)

Stereotype names are structured into a tree having **8 main entries**, which correspond to the components of our dynamic definition of "system"

It is a **generic structure** that offers a logical way to find stereotype names, no matter the domain

Just a few stereotype names are shown here

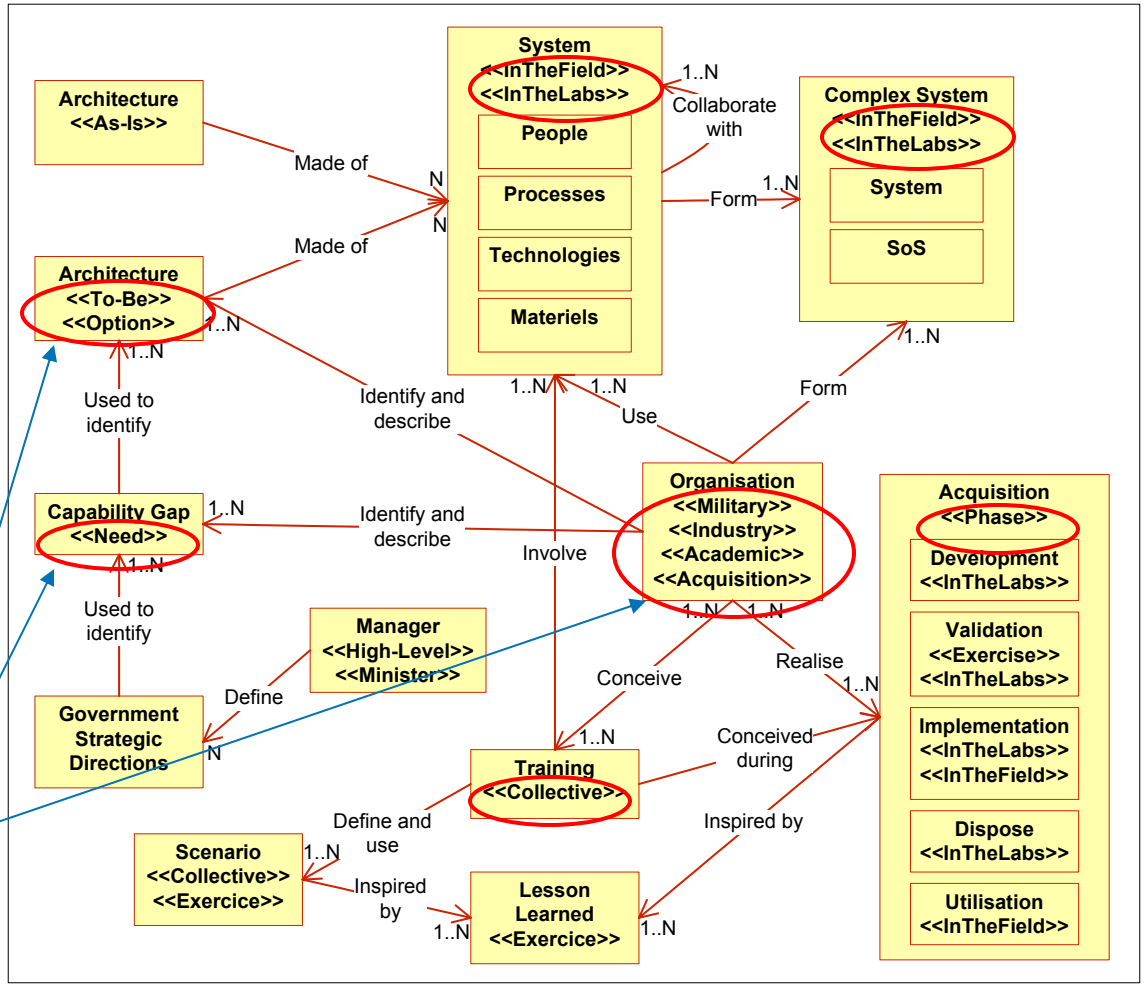


Elements of Solution (6/8)

Example 1: An over-simplified military acquisition system (class diagram: strategic view)

Organizations:

- Use systems
- Form complex systems
- Identify and describe capability gaps and architecture "to-be"
- Achieve acquisition
- Conceive training
- etc

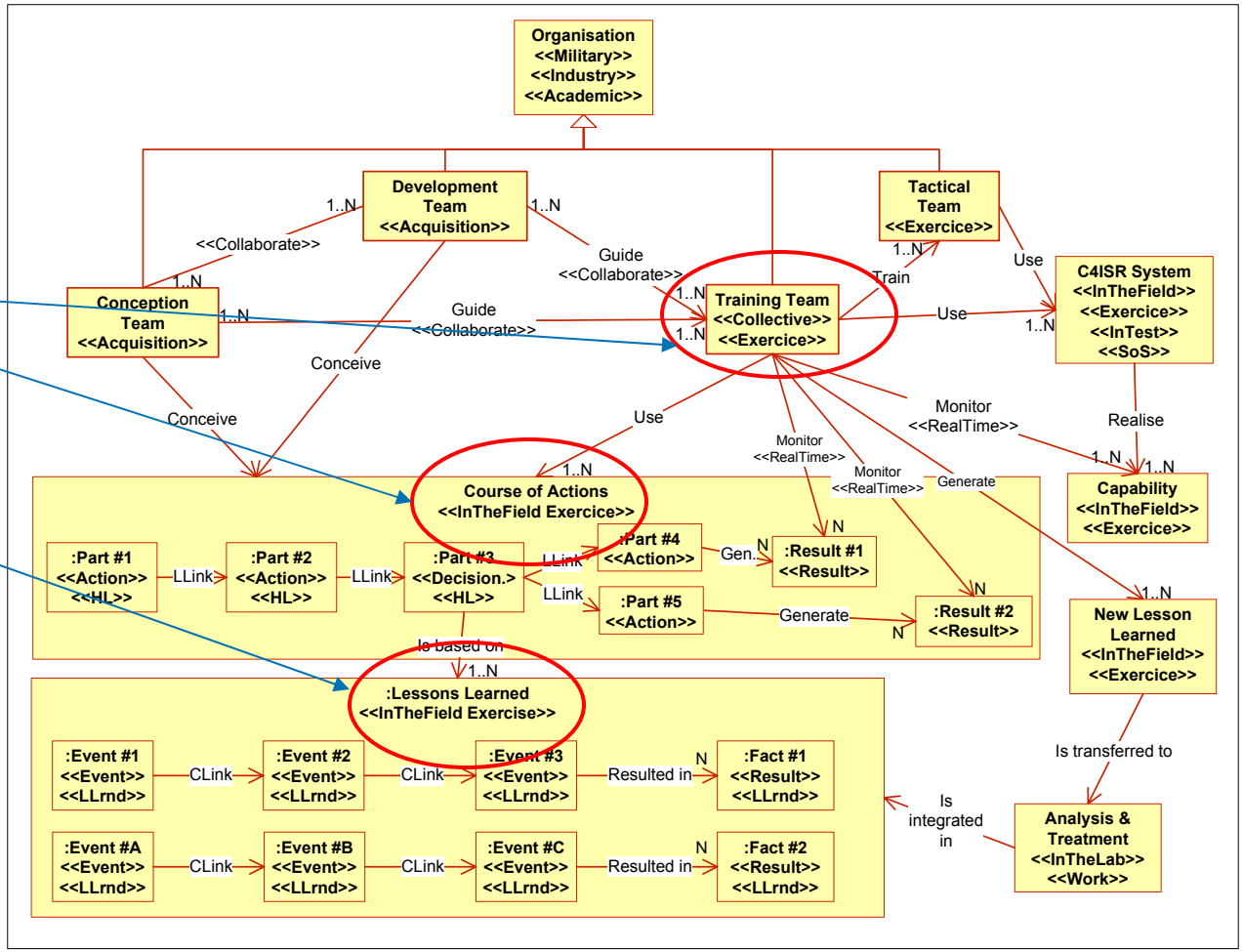


Elements of Solution (7/8)

Example 2: A over-simplified collective training "architecture" (class diagram: strategic and operational view)

Linking elements of different nature in the DB:

"**Training Team**" is linked to "**Course of Actions**" which is indirectly linked to "**Lessons Learned**"



Elements of Solution (8/8)

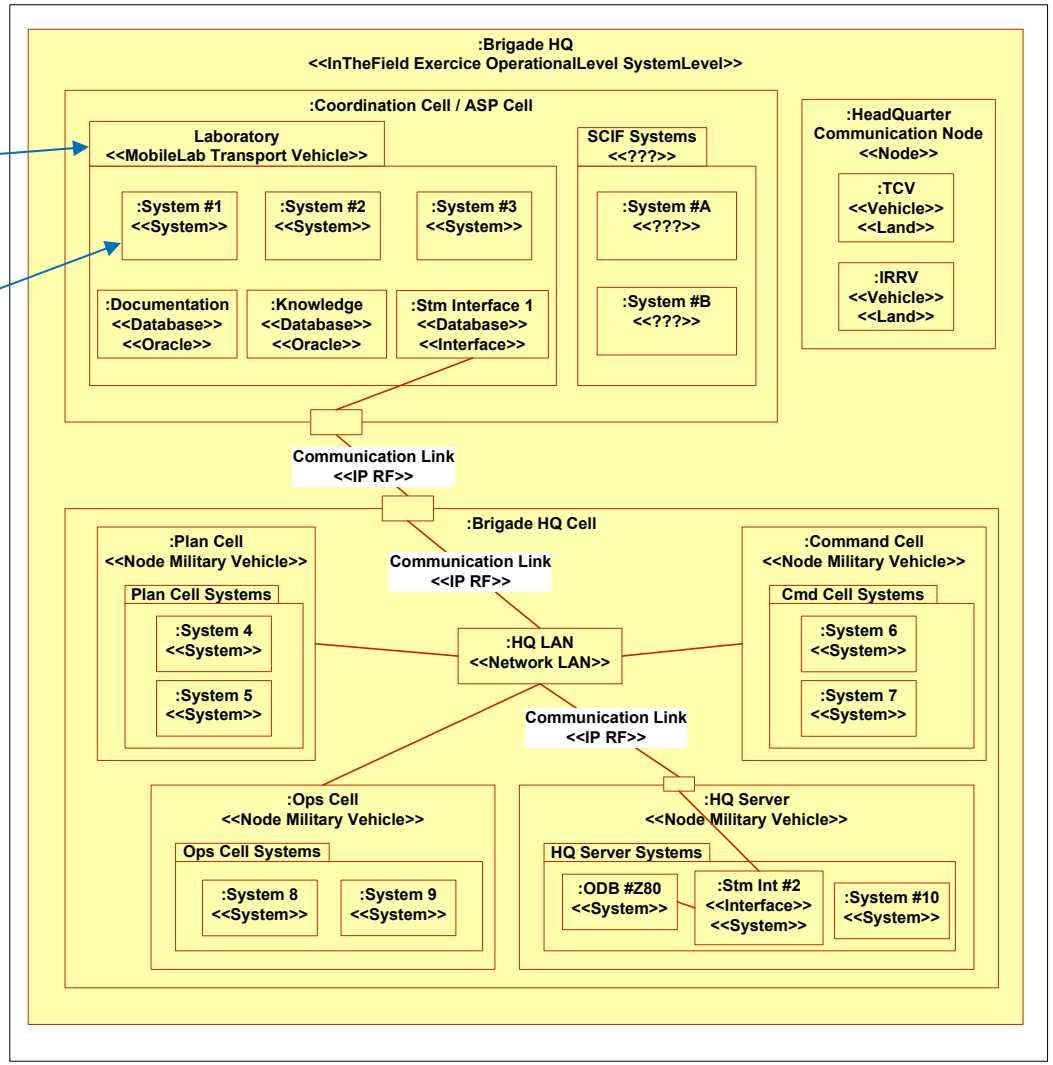
Example 3: An over-simplified C4ISR "architecture" (class diagram: operational and system view)

Modeling military materiel:

→**Packages** are used to group physical systems

→**Classes** may contain other classes (it should be possible to drill down into classes)

→Strategic, operational and tactical model elements should be linked into the DB (**holism**). Any model element that are logically linked in real life should be linked in the database



Conclusion

- The **MAU-Profile** is still under development, it remains to be validated and tested. More concrete examples are needed
- If used with appropriate CASE tools and DB:
  - It will favor holism
  - It will allow the sharing of relevant information and ease collaboration among stakeholders (no matter their domain of operation)
  - This will contribute to make the whole enterprise architecture **updated**, **synchronous,** and **homogeneous** (avoiding stovepipe projects like in the old threat-based planning)
- Having a holistic and dynamic description of the whole architecture will contribute to ease the understanding of new associated complexities (by using specialized tools like M&S for instance)