Using Relax Operators into an MDE Security Requirement Elicitation Process for Systems of Systems

Nicolas BELLOIR[1], Vanea CHIPRIANOV[1], Manzoor AHMAD[1], Manuel MUNIER[1], Laurent GALLON[1], Jean-Michel BRUEL[2]

(1) LIUPPA, University of Pau, France

(2) CNRS/IRIT Toulouse University, France

Agenda

- Challenges in Security Requirement Elicitation for SoS
- Introduction to the RELAX RE language
- Maritime safety and security case study
- An MDE-based process
- Conclusions and Perspectives

Challenges in Security Requirement Elicitation for SoS

- SoS characteristics :
  - Operational and managerial independence of composing systems
  - Evolutionary development
  - Emergent behaviour
  - Geographic distribution

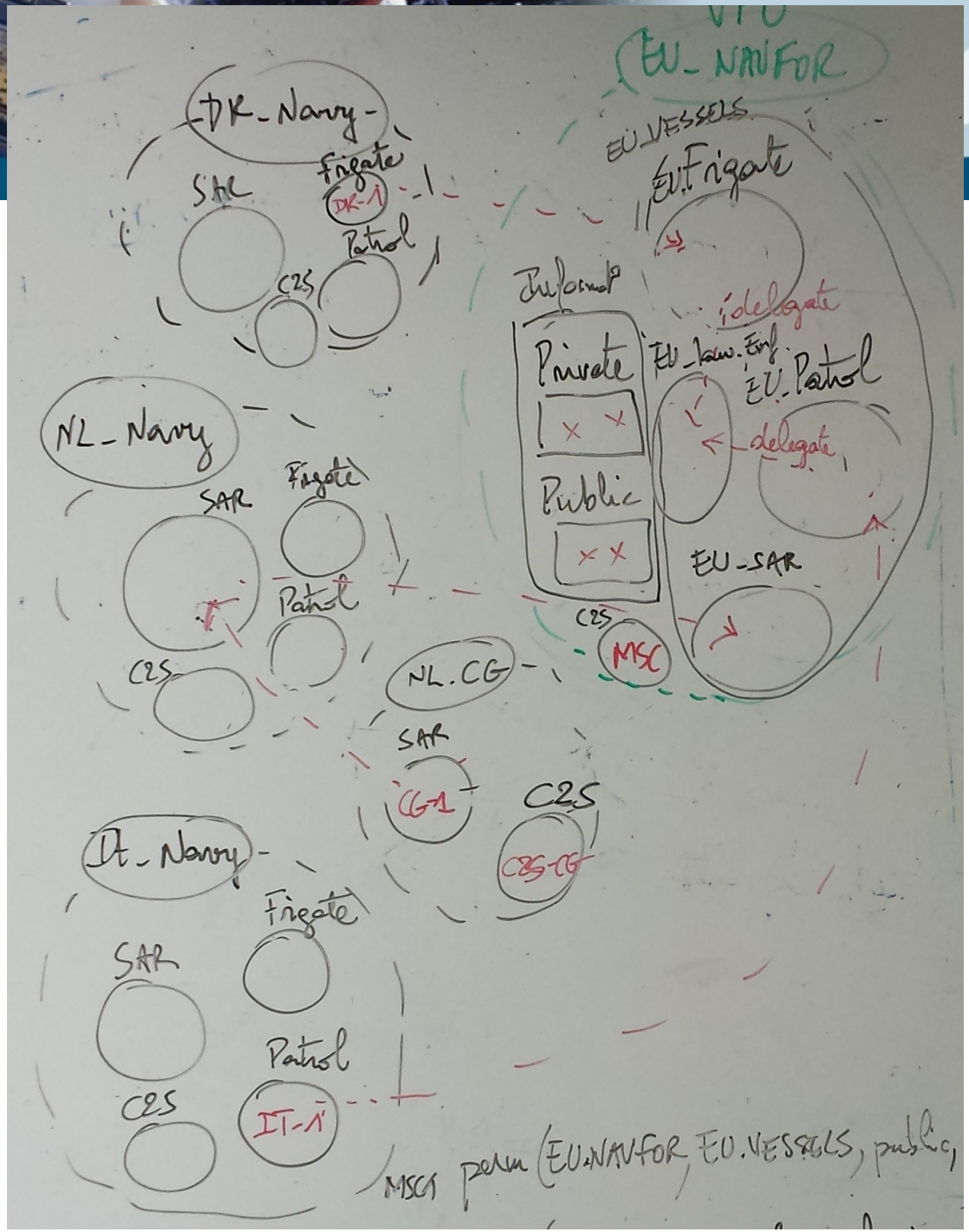
Challenges in Security Requirement Elicitation for SoS

- Security of SoS
  - Vulnerabilities of one composing system are cascaded into other systems composing the SoS
  - How to identify overarching SoS security requirements ?
  - How can security reqs be modelled so as to integrate them into functional reqs modelling ?
  - How to identify and allocate reqs to composing systems for their respective teams to manage?

Introduction to the RELAX RE language

- Types of requirements :
  - Invariant : SHALL
  - Relaxed : MAY - reqs that could temporarily be modified under certain conditions
    - ENV : operating context of the system
    - MON : *observable* properties of the context
    - REL : in what way the observable can be used to derive info about the context
    - DEP : impact on dependent reqs of the relaxed reqs

Maritime safety and security case study*



* inspired from [17]

Maritime safety and security case study

- EU_NAVFOR – SoS
  - EU_Law_enforcement = ships which, at a certain moment, have the task of preventing/figthing crime
  - Information :
    - Public
    - Private
  - MSC=European C2S, verify rigths to access information

Maritime safety and security case study

- Textual security reqs
  - Msc1 : Operators on vessels of the EU_NAVFOR can access public information about the ships transiting in the operation area.
  - Msc2: Operators on vessels of the EU_NAVFOR which are assigned to the prevention of criminal activities (or similar tasks) can access additional “off the record“ information about ships which has been gathered during the operation.
  - Msc3 : Operators on SAR vessels certified by EU_NAVFOR members can access all the information about a ship in case of emergency.

Maritime safety and security case study

- **Security reqs modelled in OrBAC :**
  - ***Rule : predicate(organisation, role, action, resource, context);***
  - **Msc1 : permission(EU_NAVFOR, EU_Vessels, read_info, public_info, default_context);**
  - **Msc1-2 : prohibition(EU_NAVFOR, EU_Vessels, read_info, private_info, default_context);**
  - **Msc2 : permission(EU_NAVFOR, EU_Law enforcement, read_info, private_info, default_context);**
  - **Msc3 : permission(EU_NAVFOR, EU_SAR, read_info, all_info, emergency);**

Maritime safety and security case study

- **OrBAC conflicts**

Abstract conflicts | Concrete conflicts | Separation constraints | Rules priorities

update

Rule name	Type	Organization	Role	Activity	View	Context
MSC3	permission	EU_NAVFOR	EU_SAR	read_information	Information_on_t...	Emergency
MSC1-2	prohibition	EU_NAVFOR	EU_VESSELS	read_information	off_the_record_i...	default_context
MSC2	permission	EU_NAVFOR	EU_Law_Enforce...	read_information	off_the_record_i...	default_context
MSC1-2	prohibition	EU_NAVFOR	EU_VESSELS	read_information	off_the_record_i...	default_context

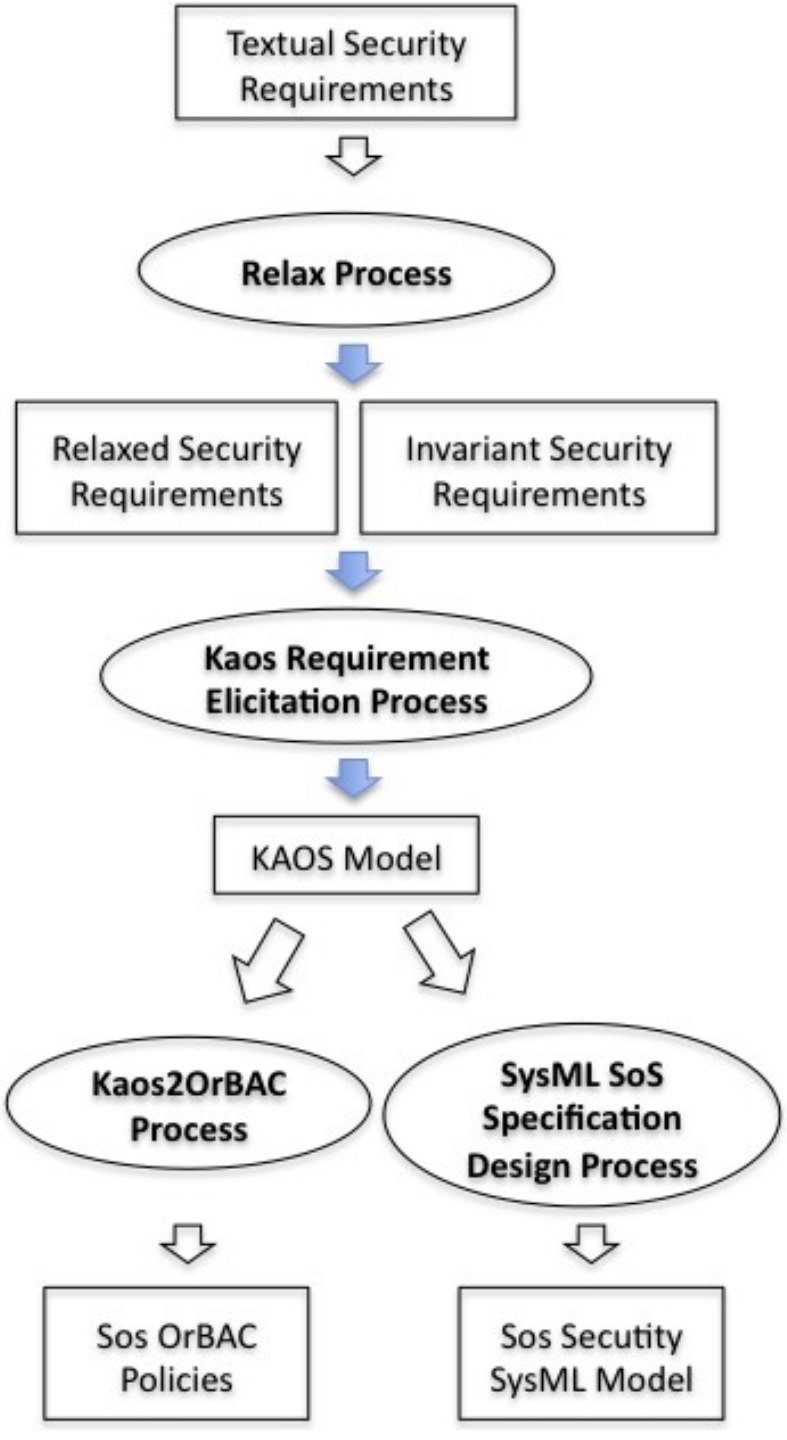
Maritime safety and security case study

- Relaxing security reqs to limit their conflicts
  - Relaxed Msc2 and Msc3 :
    - Private information MAY be read by ships that are executing a task of fighting against crime OR by SAR ships in case of emergency.
      - ENV : fight against crime (FAC), access to private information (API)
      - MON : Aggression level (AL), Access rules (AR)
      - REL : FAC = (AL > 10 ? true; false); API = select ∗ from AR where . . .
      - DEP : it has a positive dependency on Msc1-2.

Maritime safety and security case study

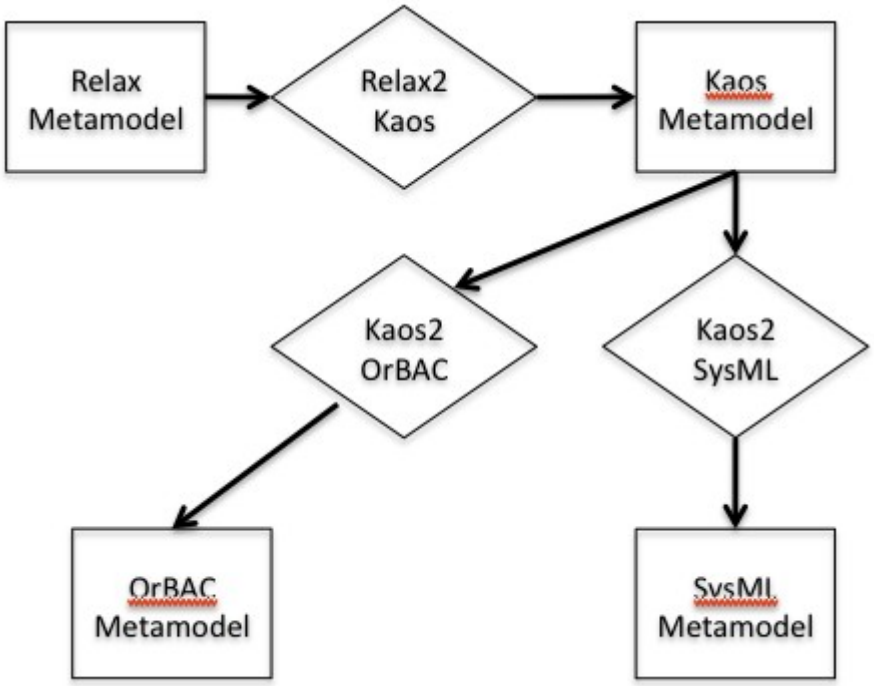
- OrBAC verification of relaxed reqs
  - Todate, there is no OrBAC operators/predicates to model the RELAX operators of MAY, OR
  - => no formal proof there is no more conflict, just intuitively

An MDE-based process



An MDE-based process

- Metamodels and Model Transformations chain



Conclusions and Perspectives

- Conclusions
  - Process for security reqs of SoS
  - Enables identifying conflicting rules early in the development cycle
- Perspectives
  - Mutual enrichment of RELAX and OrBAC :
    - Add to RELAX operators to make the difference between *context* and *role*
    - Add to OrBAC concepts to account for RELAX operators SHALL, MAY, OR, AND