De KAOS à RBAC :
conception de règles de contrôle d'accès à partir d'une analyse des besoins - une étude de cas

From KAOS to RBAC:
a case study in designing access control rules from a requirements analysis

Yves Ledru, Jean-Luc Richier, Akram Idani, Mohamed Amine Labiadh

Université Grenoble Alpes/ Grenoble INP /CNRS

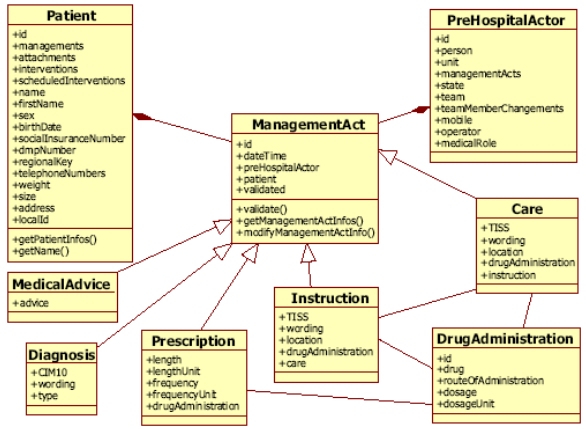
Laboratoire d'Informatique de Grenoble

6th conf. On Network Architectures and Information Systems Security, La Rochelle, May 2011
This work is sponsored by the ANR Selkis Project (ANR-08-SEGI-018)
and MODMED Project (ANR-15-CE25-0010)

Goal of this research

- Elicit requirements for a security policy
- Design a set of access control rules enforcing the policy.

- Our starting point: a set of UML diagrams specifying the functional aspects of the Information System (IS)
  - UML class diagrams
  - UML use cases

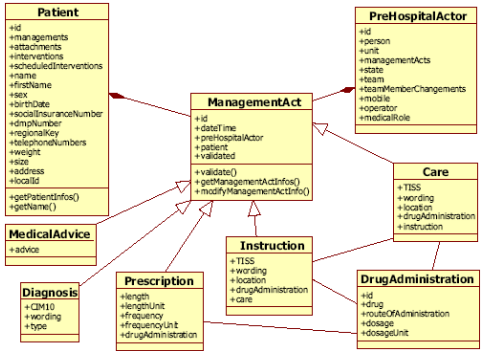


Our case study : Res@mu

- Information system for an urgency medical help service (SAMU)

SAMU

- Developed by IFREMMONT, a french association for e-medecine.



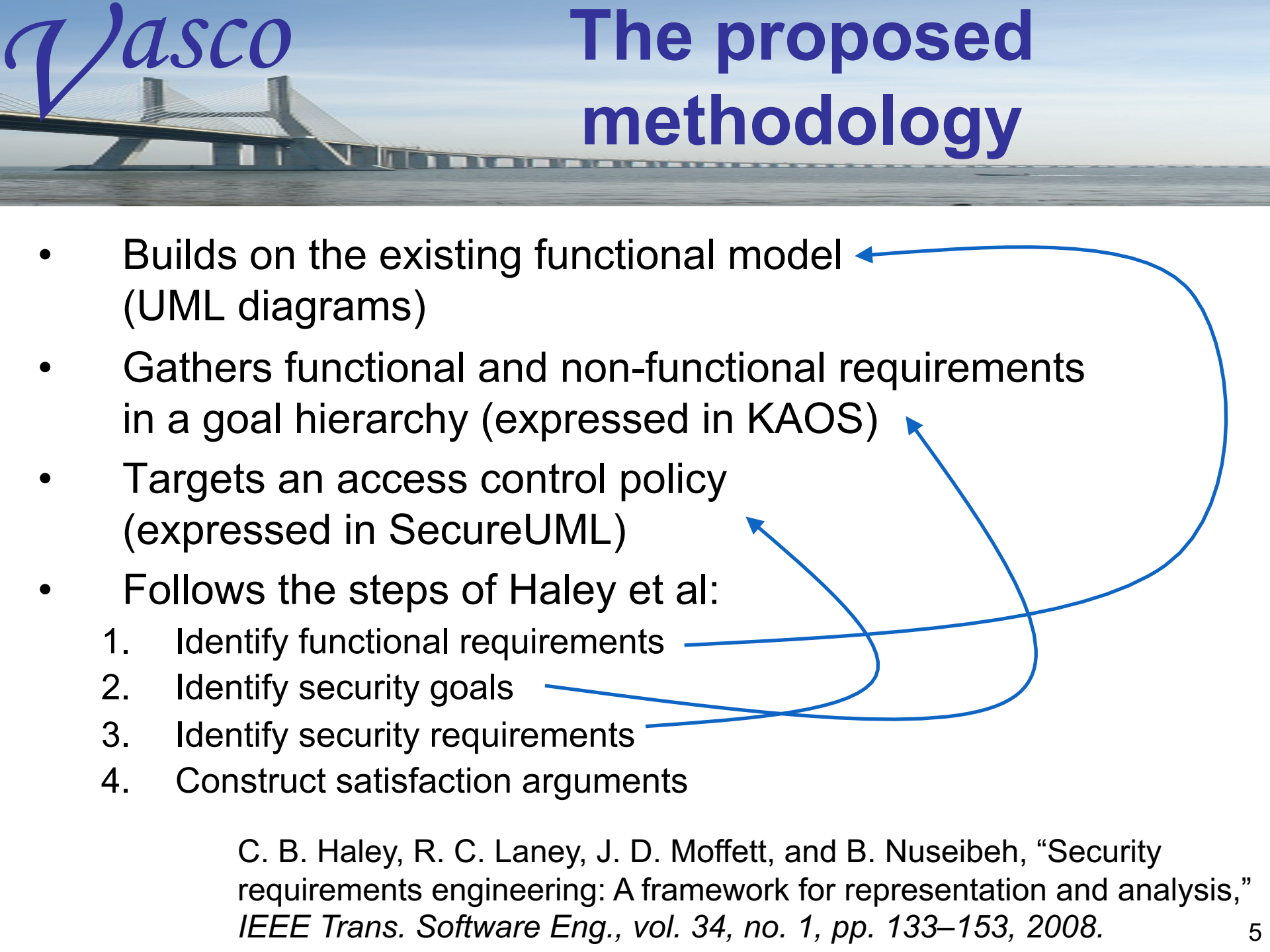
- Functional model :
  77 classes, 100 use cases developed before this study.

The need for security in Res@mu

- Access to the information system must be restricted to authorized personal
- The authorized personal are numerous and evolve over the life-time of the information system => need for a role-based approach
- Medical data
  - Are confidential
  - Must be available to the rescue teams
  - Must be protected against unauthorized modifications (integrity)

The proposed methodology

- Builds on the existing functional model (UML diagrams)
- Gathers functional and non-functional requirements in a goal hierarchy (expressed in KAOS)
- Targets an access control policy (expressed in SecureUML)
- Follows the steps of Haley et al:
  1. Identify functional requirements
  2. Identify security goals
  3. Identify security requirements
  4. Construct satisfaction arguments



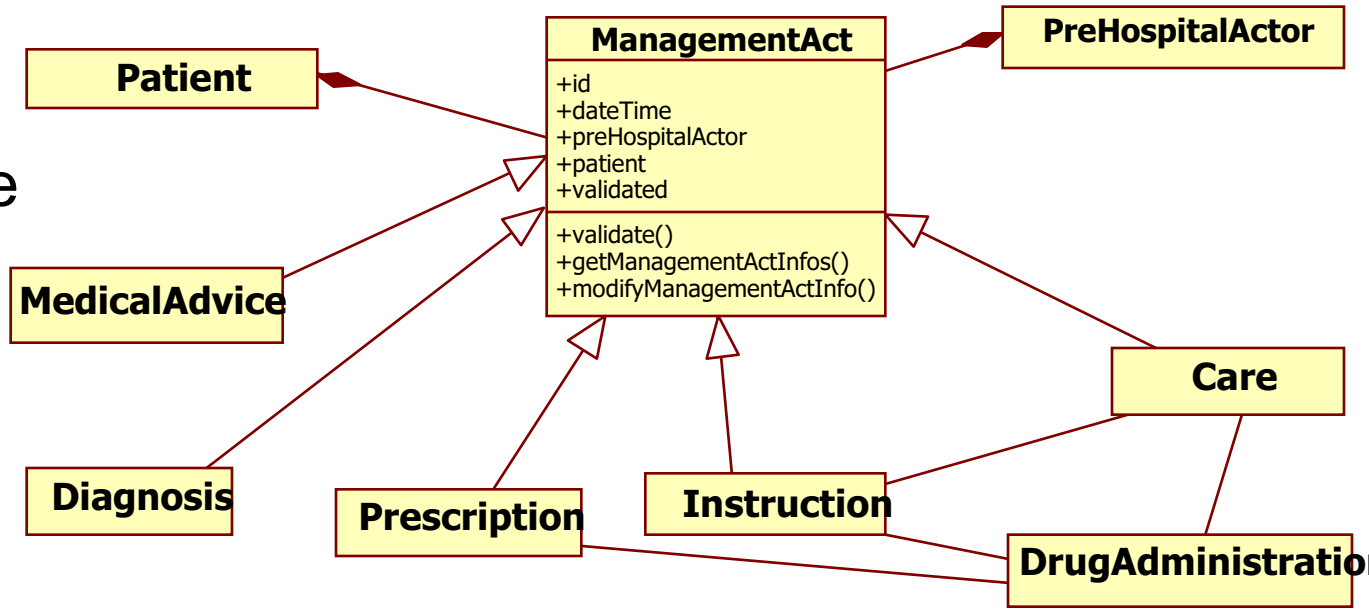
C. B. Haley, R. C. Laney, J. D. Moffett, and B. Nuseibeh, “Security requirements engineering: A framework for representation and analysis,” *IEEE Trans. Software Eng., vol. 34, no. 1, pp. 133–153, 2008.*

The proposed methodology : 8 steps

0. (Build a functional model)
1. Construct agent hierarchy
2. Identify relevant use cases
3. Construct a functional goal hierarchy
4. Identify security goals
5. Refine into security requirements
6. Design RBAC rules
7. Check satisfiability of functional goals

0. Functional model

- Security target : ManagementAct
- Management acts are all kinds of medical acts (diagnosis, advice, prescription, instruction, care,…)
- Each medical act is linked to a single patient
- Prehospital actors are the medical personal who perform these acts (they must be qualified for the medical act)

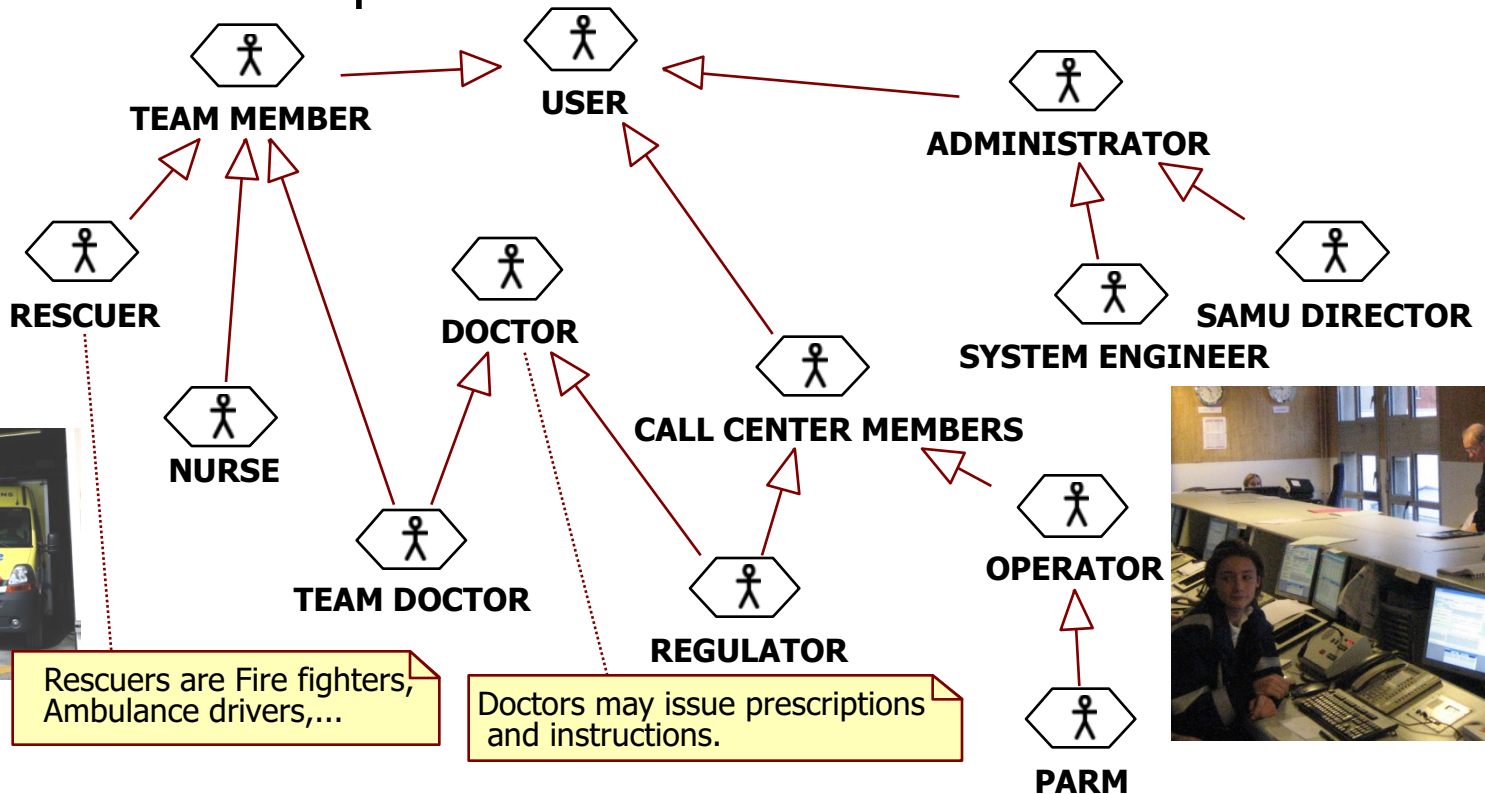


1. Construct an agent hierarchy

- KAOS associates goals to agents responsible for these goals
- Agents were identified during long discussions, based on the presentation of the functional model by the IFREMMONT domain experts
- KAOS agents are candidates for RBAC roles.

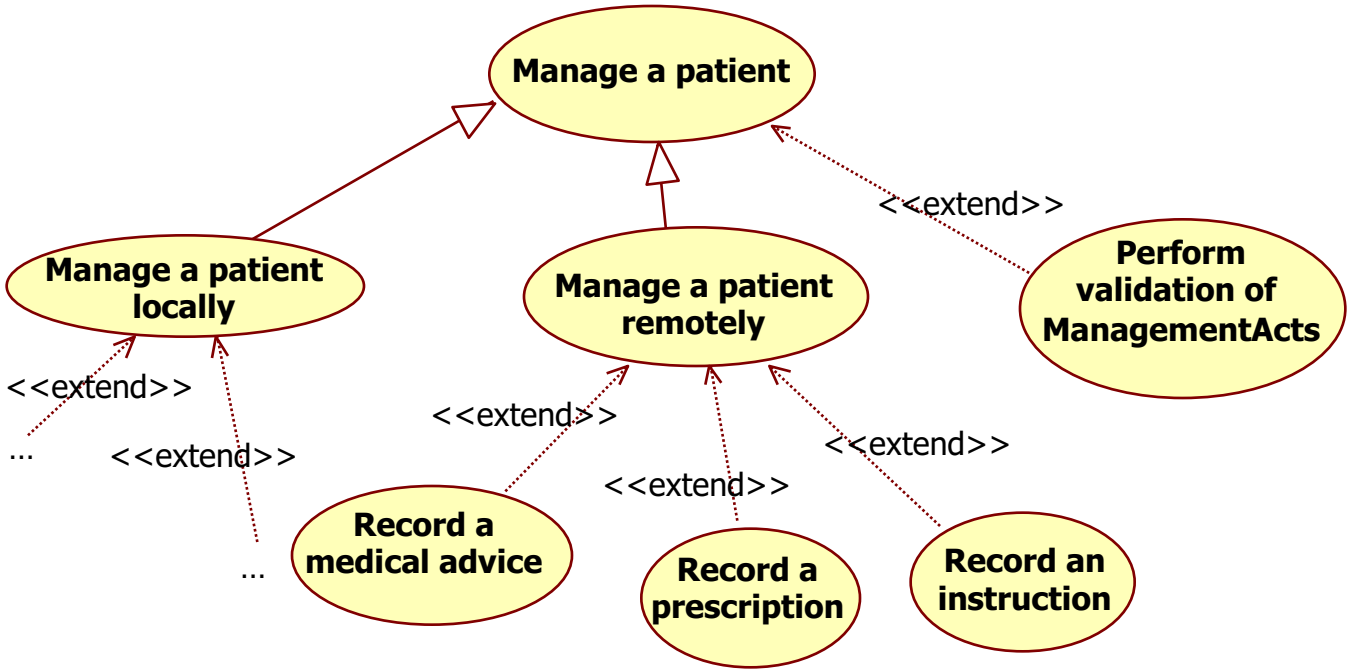
1. Construct an agent hierarchy (2)

- The agent hierarchy distinguishes between
  - Doctors vs non doctors
  - Mobile team members, call center members and administrative personal

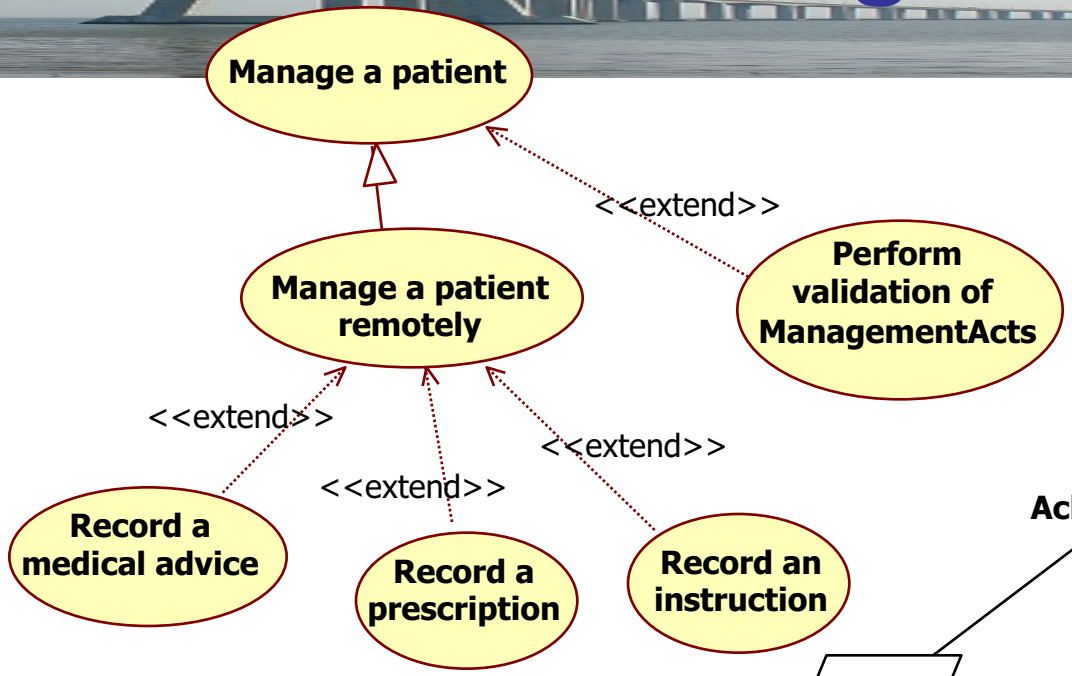


2. Identify relevant use cases

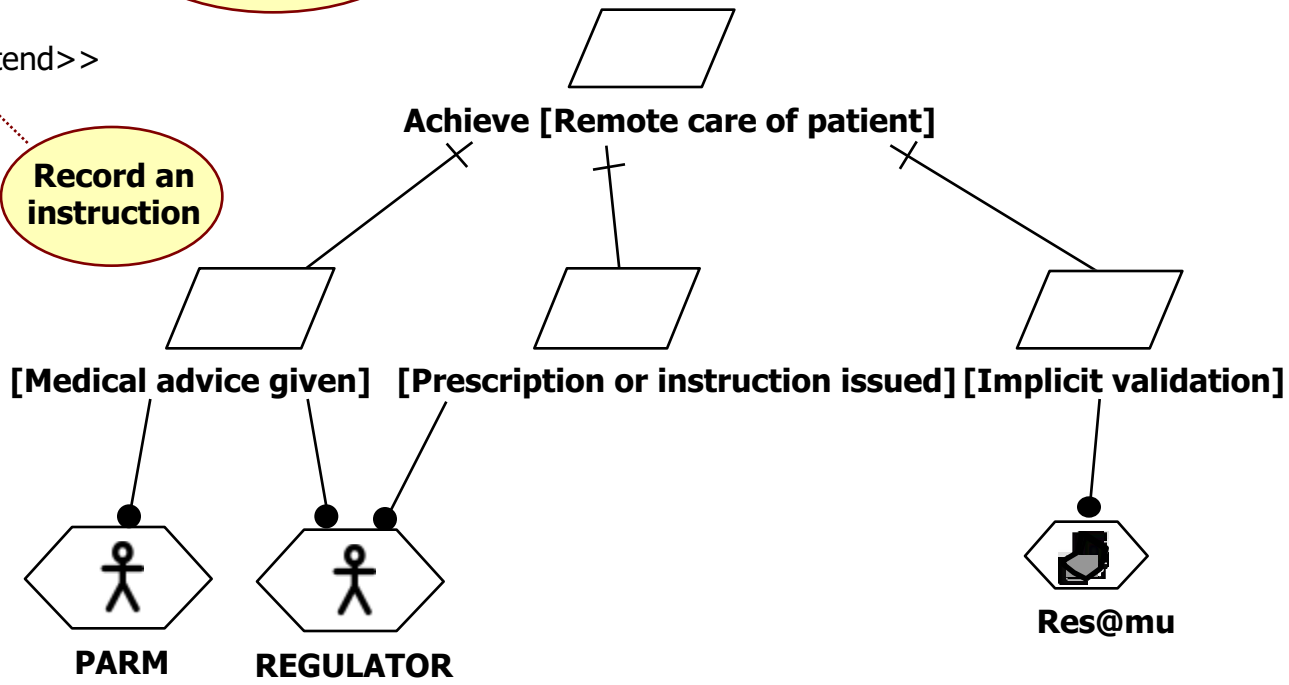
- Keep only the use cases relevant to management acts (28 out of 100)
- For example, the following use cases detail remote and local management of a patient
- In both cases, acts must be validated!



3. Construct a functional goal hierarchy



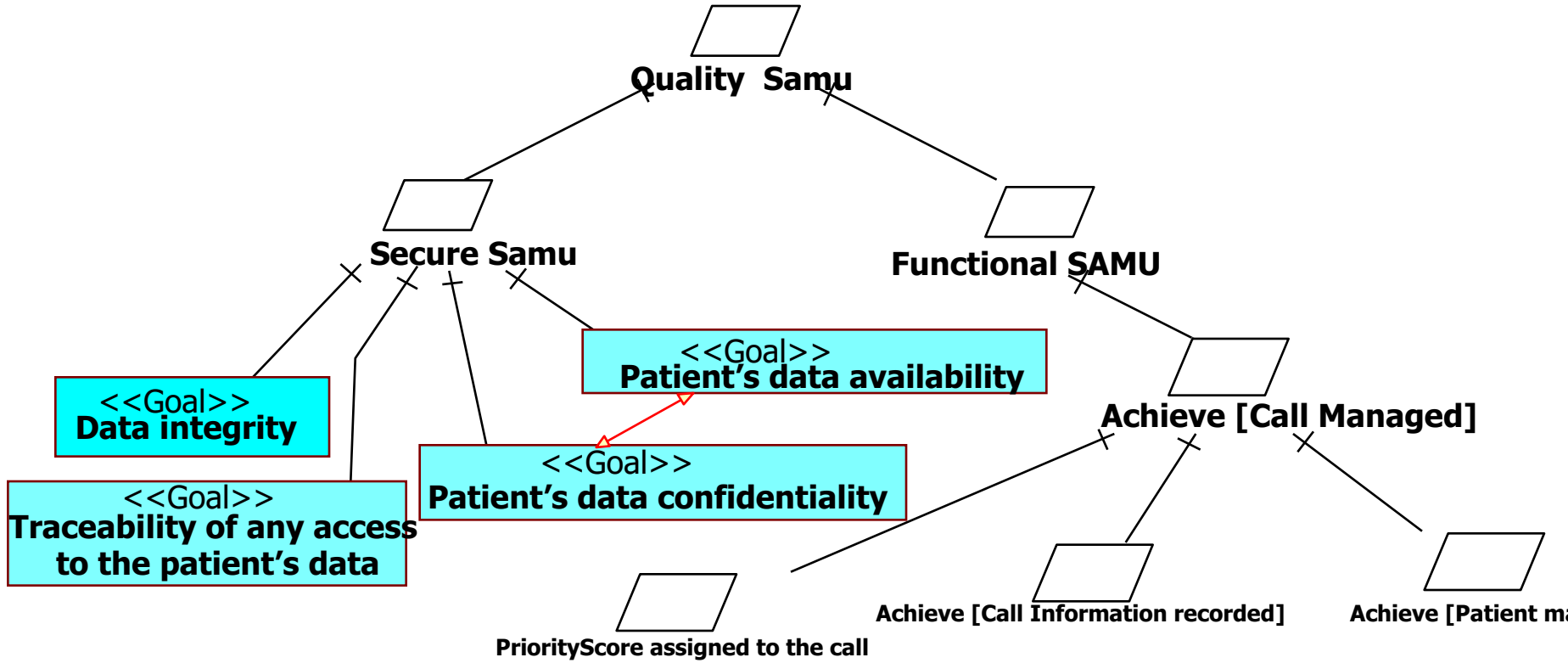
- Use cases are turned into KAOS functional goals



- Concrete goals are linked to agents (human or software)

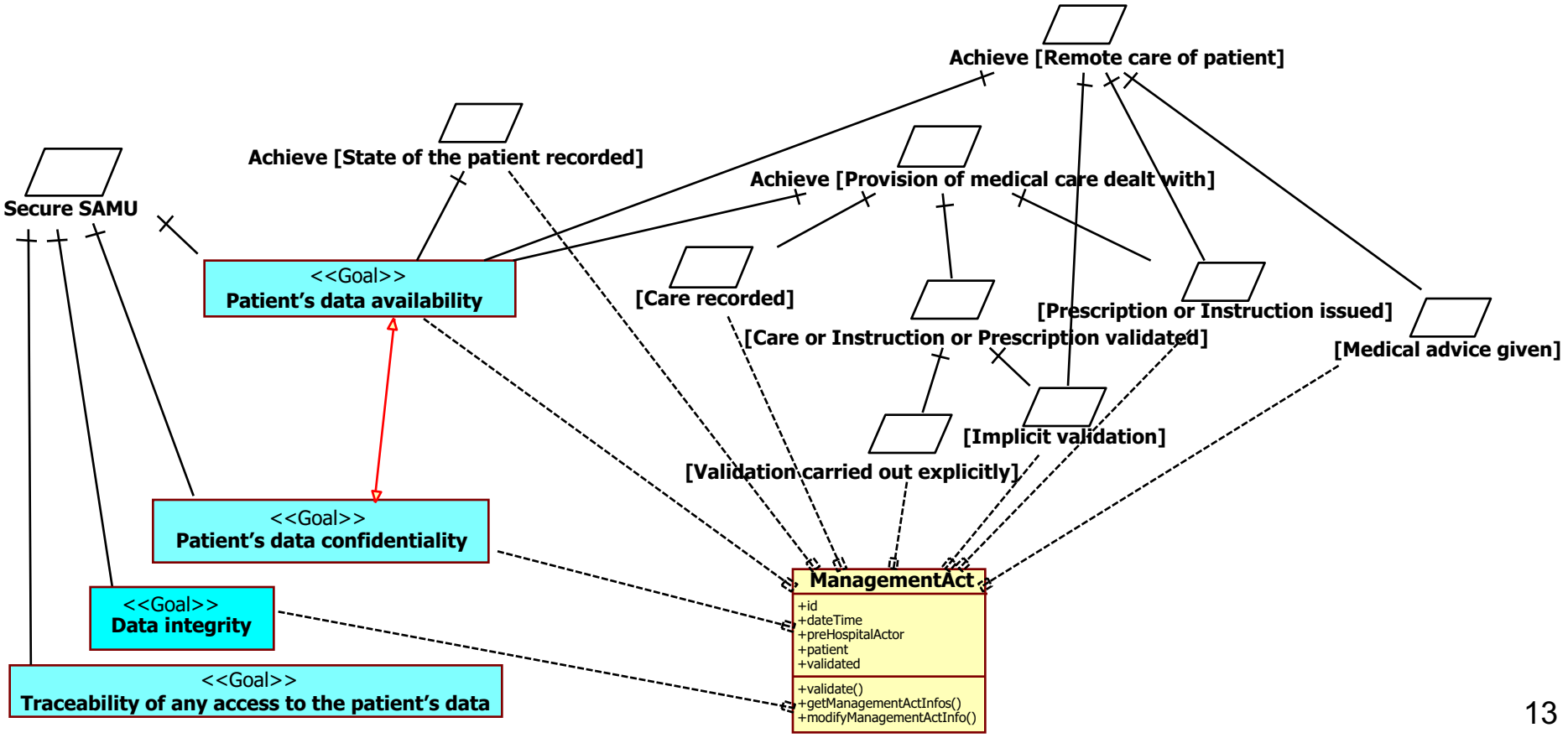
4. Identify security goals

- Security goals identified by reviewing ACIT properties (availability, confidentiality, integrity, traceability)
- This results in two goal hierarchies (functional or not)



5. Refine into security requirements

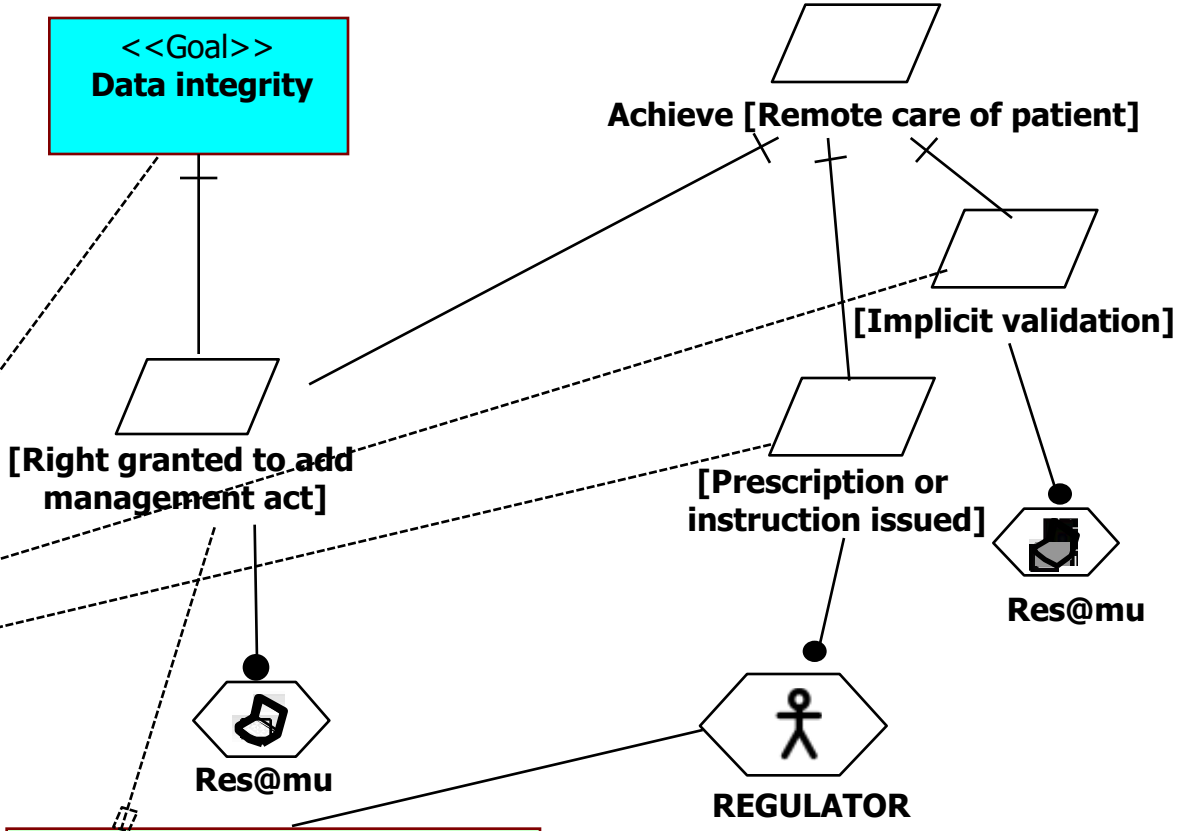
- Find all functional and non-functional goals linked to Management acts



5. Refine into security requirements

Security goals

Functional goals



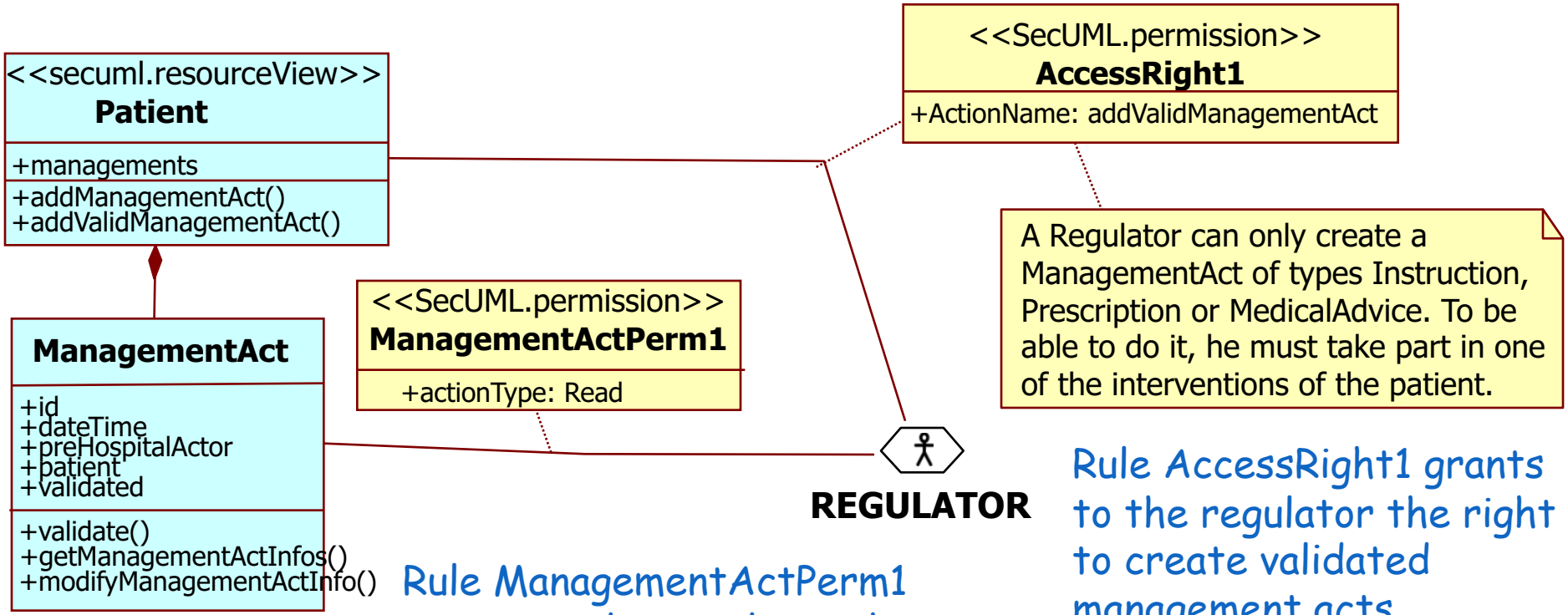
These goals are linked to the same data to protect !

An additional goal (security requirement) ensures that the security goal applies in the context of the functional goal!

The security requirement is fulfilled by an access control rule

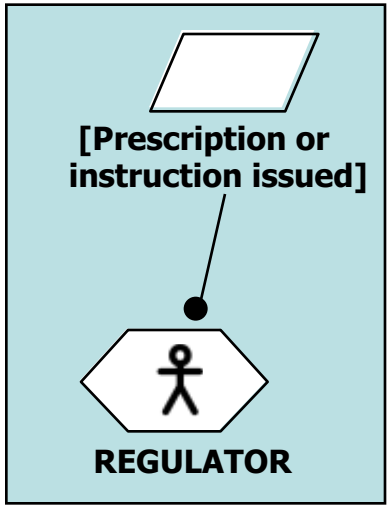
6. Design RBAC rules

- Access control rules are expressed in the SecureUML syntax



7. Check satisfiability of functional goals

- For each functional goal, check that there are sufficient permissions to authorize it!
- For example, « Prescription or Instruction Issued » requires that the regulator
  - Is allowed to create a management act of subclasses prescription and instruction (OK due to **AccessRight1**)
  - Has read access to all other management acts related to the patient associated to the prescription or instruction (to avoid interference with other current management acts)
    (OK due to **ManagementActPerm1**)



Comparison with Haley et al

- **Haley et al**
  - 4 steps
  - Based on Jackson's problem frames
  - A formal verification process based on causal logic
- **Our KAOS2RBAC approach**
  - 8 steps covering the 4 steps of Haley et al
  - Based on KAOS : a richer framework with goals linked to agents, data, permission rules
  - Our KAOS diagrams allow traceability between security goals and access control rules
  - Our verification step remains rather informal

Conclusion

- An approach to design access control rules from security goals
- Applied on a real world case study, using an existing functional model
- Provides traceability from goals to rules
- Perspectives
  - KAOS suggests the systematic identification of « obstacles » to the most concrete goals to make the model stronger
  - This identification could benefit from a risk analysis based on a list of standard attacks.

Questions?

Photo Credits:
http://commons.wikimedia.org/wiki/File:Vasco_da_Gama_Bridge_03.JPG
http://commons.wikimedia.org/wiki/File:Sala_de_Regulacion_del_Samu_de_Paris.jpg?uselang=fr
http://commons.wikimedia.org/wiki/File:Logo_Samu.gif?uselang=fr
http://commons.wikimedia.org/wiki/File:H%C3%B4pital_d%27Orl%C3%A9ans-la-Source_SAMU_1.jpg?uselang=fr