APCCM2019

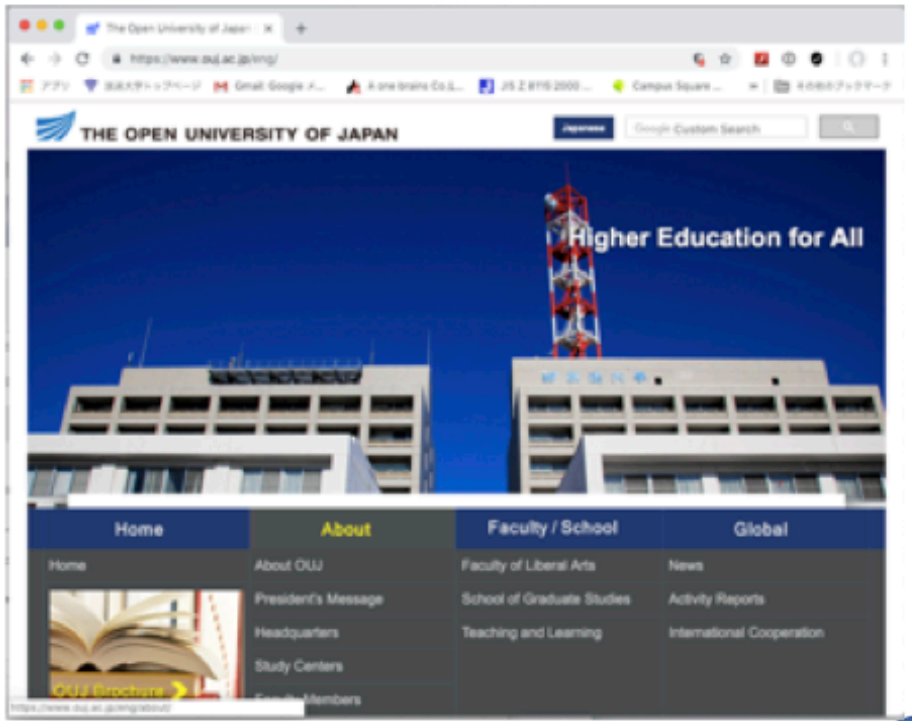
A method to generate traverse paths for eliciting missing requirements

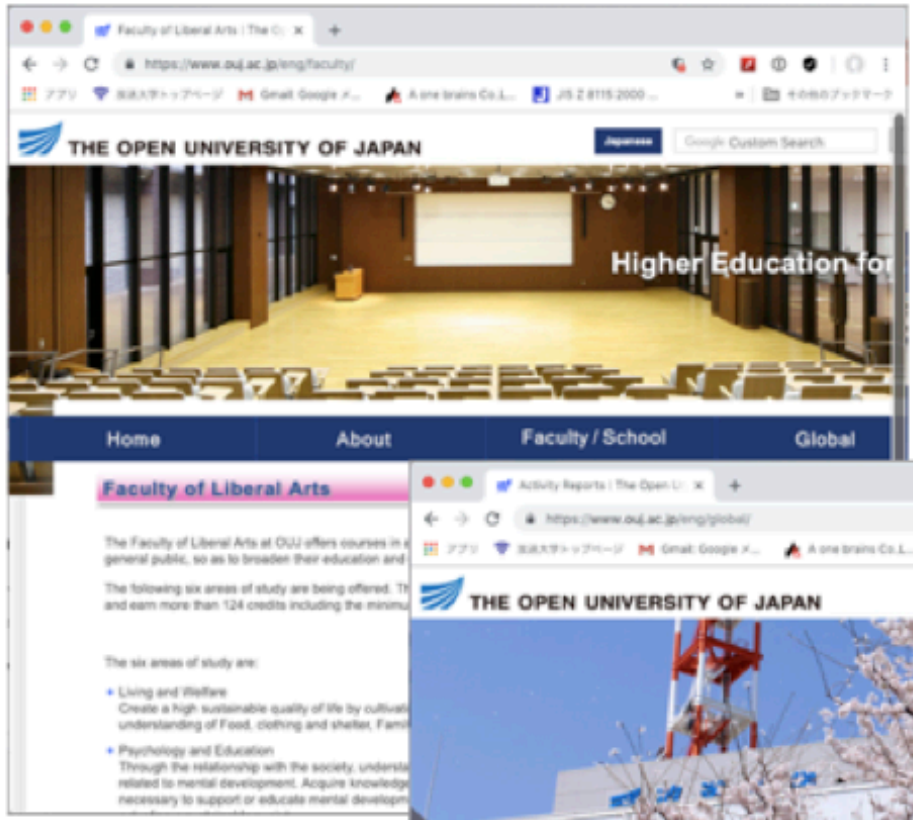
Takako Nakatani (The Open University of Japan)
Hideo Goto (National Institute of Informatics)
Taichi Nakamura (National Institute of Informatics)
Osamu Shigo (Tokyo Dennki University)

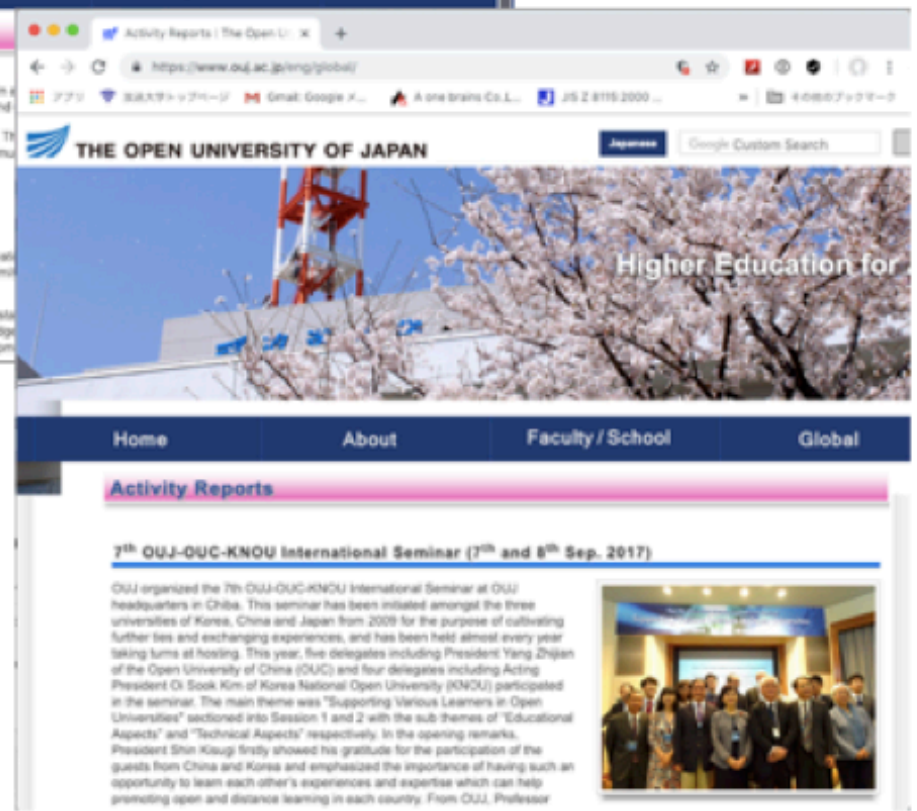
Agenda

1. Background and a goal
2. Related work
3. Approach
4. Overview of the proposing method
5. Experiment
6. Discussion
7. Conclusion

1. Background







Goal

- To save people from getting lost in a web, we develop a method to create an ”adequate/useful” web site based on a conceptual model.
- The mission of the method
  - Define elements of a conceptual model to achieve the goal.
  - Define a process to find requirements that are not satisfied in the current system.
  - Generate actors’ traverse paths within the system.
    - Ex. of a traverse path) Actor *A* accesses *X*, *Y* from *X*, and *Z* from *Y*.
  - Generate inverse requirements.
    - Ex.) Actor *A* cannot access *P* from *X*.

2. Related work

- Use case as a requirements definition method.
  - We need to represent processes of users' total activities/behaiviours in the developing web system.
- Ontology as a requirements elicitation method.
  - We focus on relations among objects with accessing permissions of each user.
- Persona analysis as a requirements engineering method to derive typical user's requirements.
  - We need more general model. -> A conceptual model.

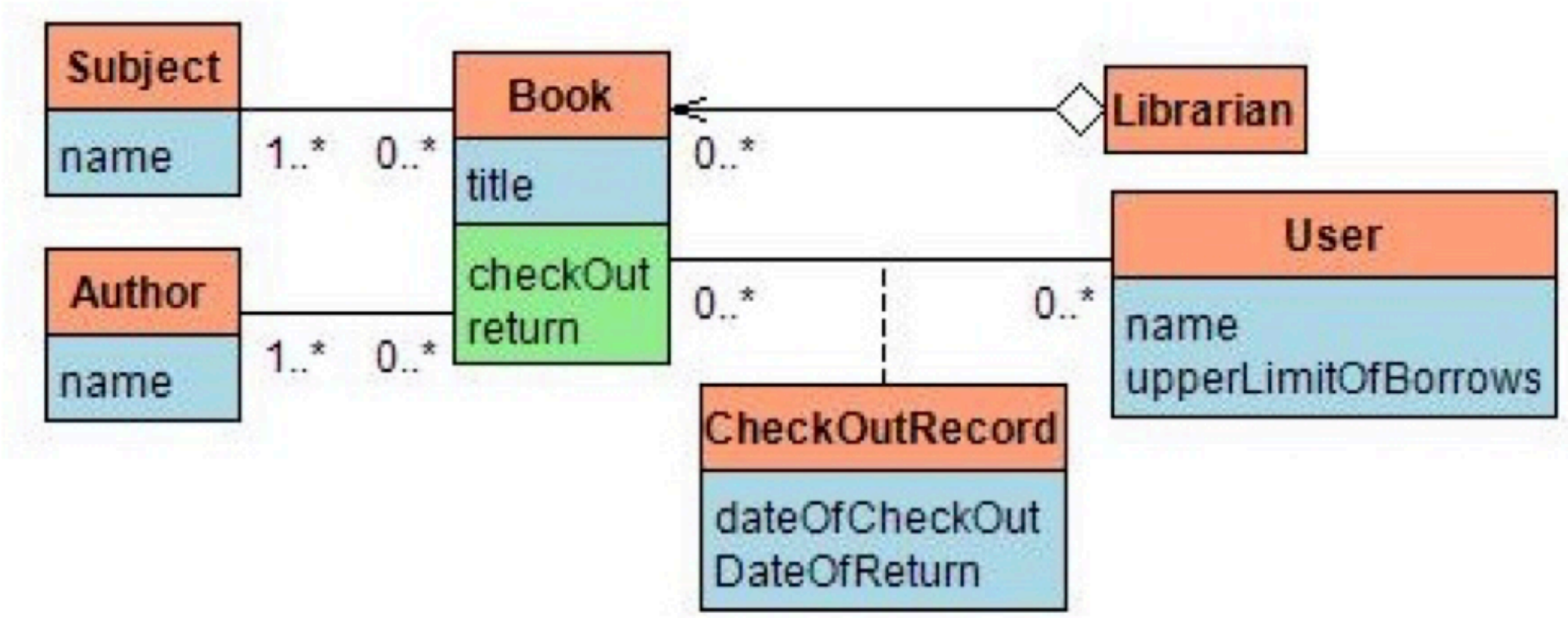
3. Approach

We extend
UML class diagram.

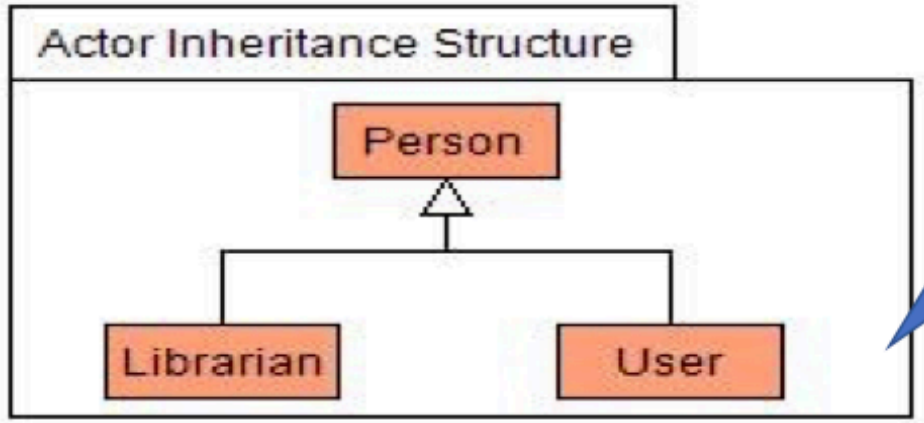
- Defining elements of a conceptual model
  - Object / Class / Inheritance
  - Association / Aggregation / Association class
  - Actor
  - Permission (actor + access permissions (create/read/update/delete))
- Defining rules for generating traverse paths
  - depending on the structure of the conceptual model.
- Applying the method and tool for an example to evaluate the method.

Overview: How does our method work?

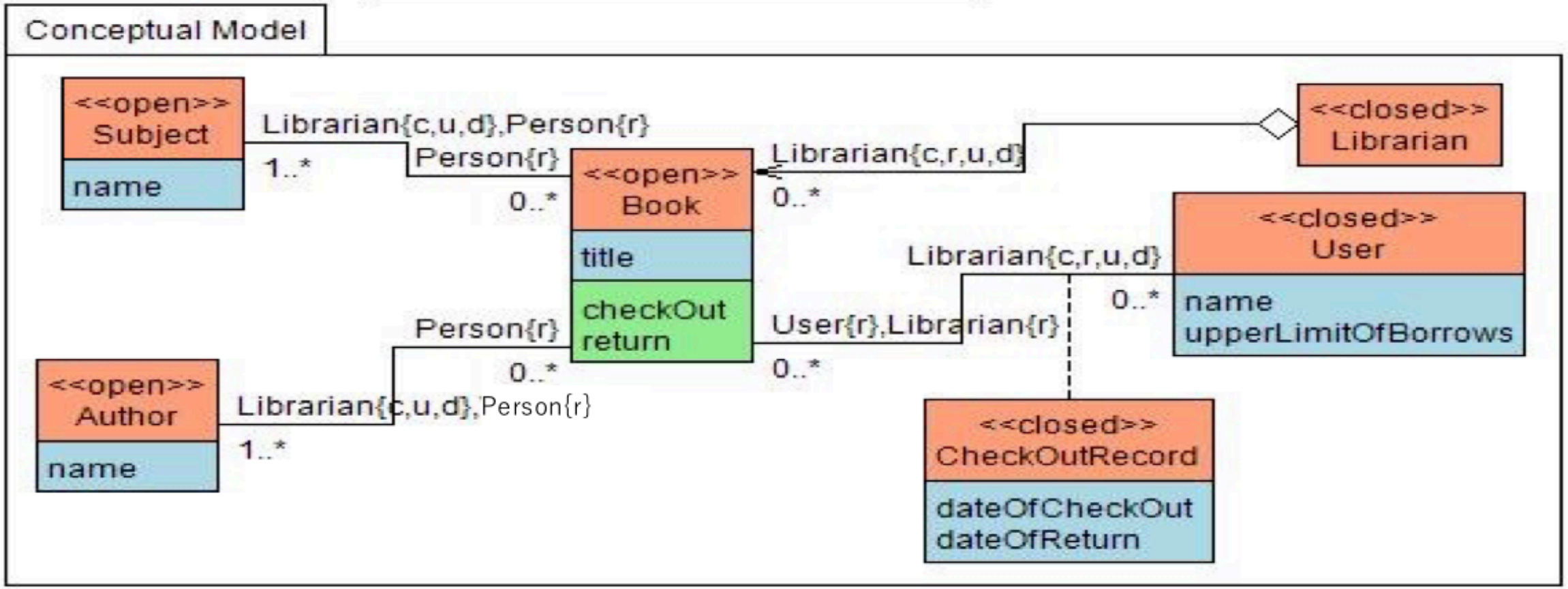
- A general class diagram does not work.



It works!

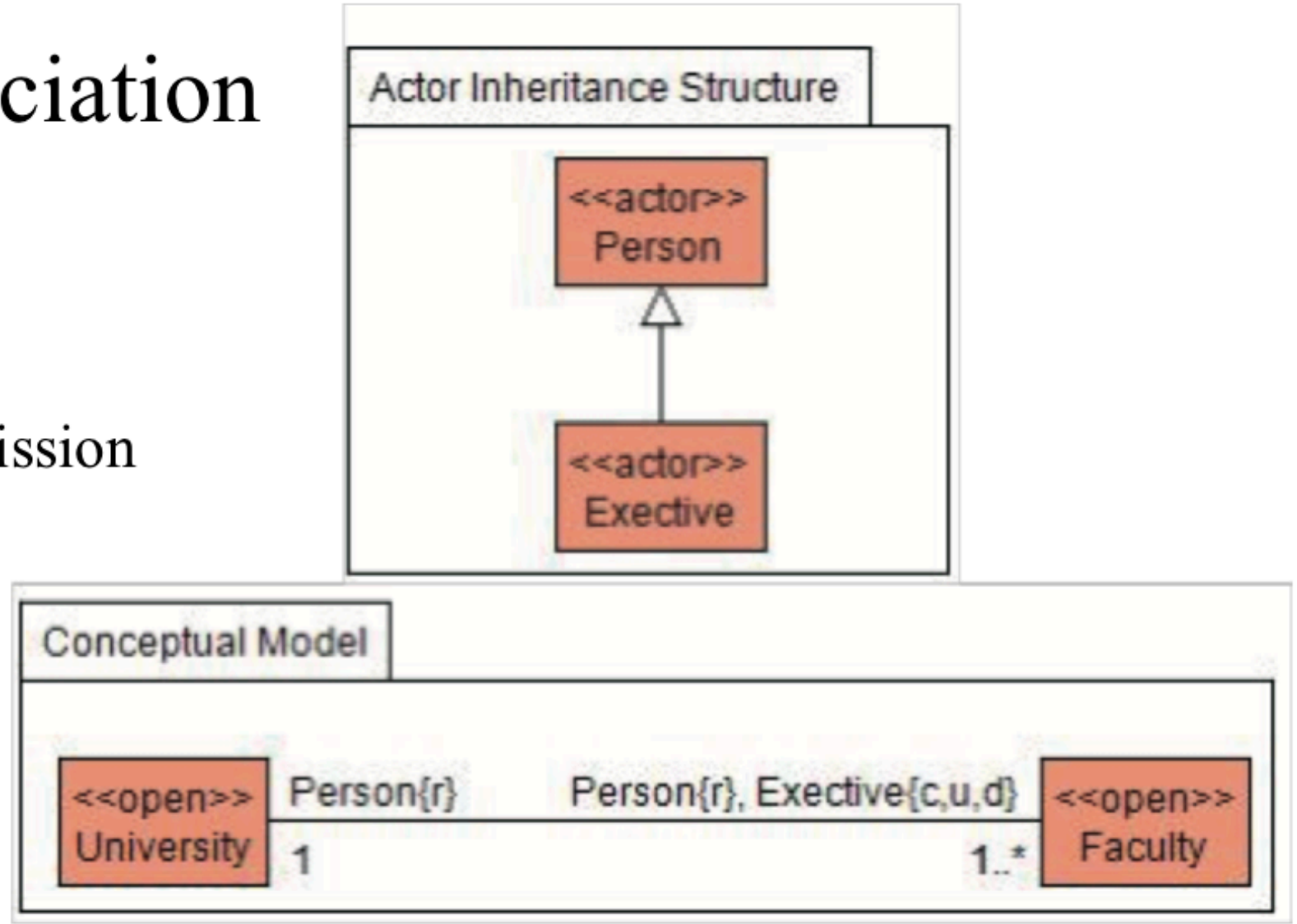


Proposed class diagrams with access permissions.



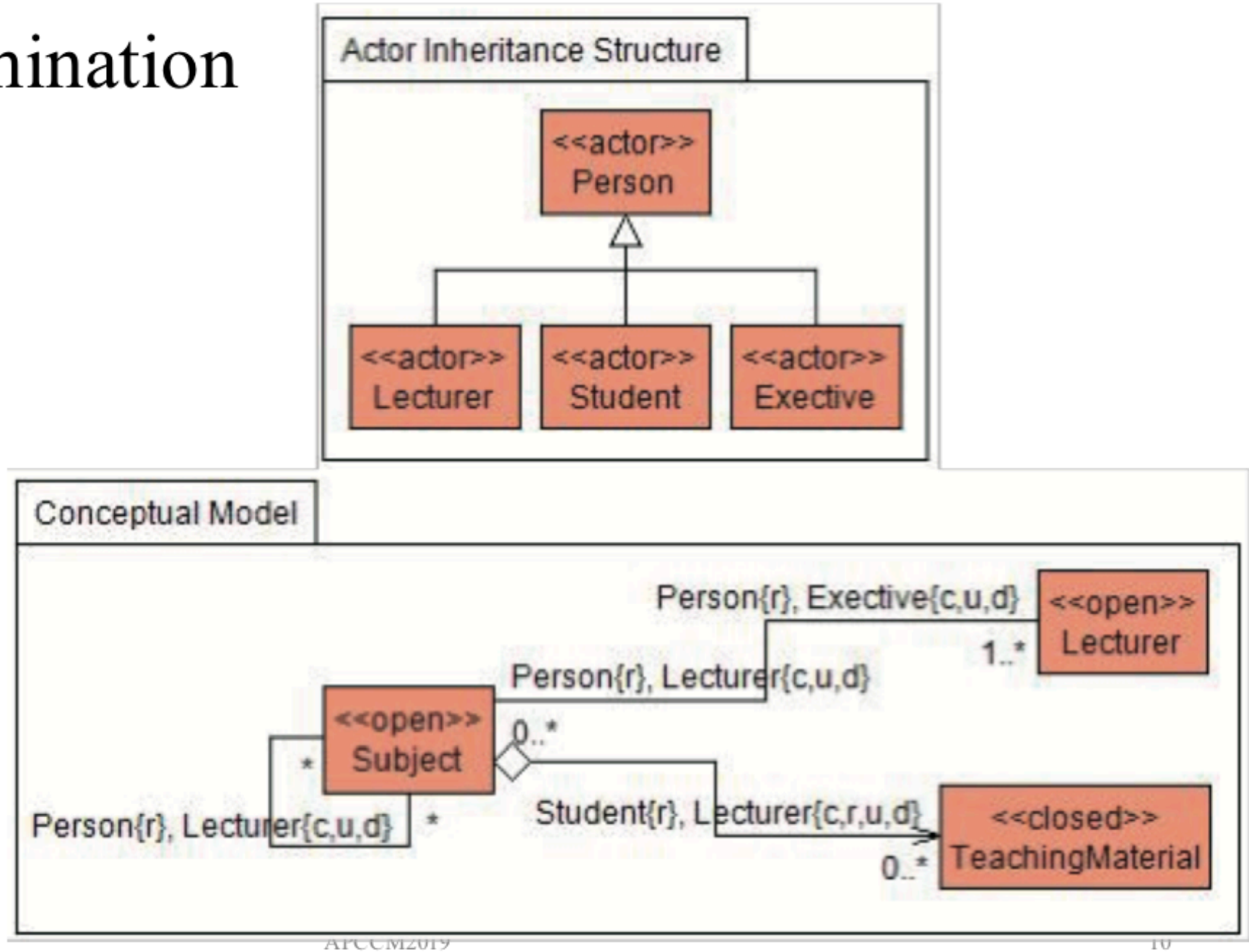
Rules for association

- <<open>>
- <<closed>>
- roles -> access permission



The tool is available from http://www.s-lagoon.co.jp/Traverser

Rules of termination of traverses



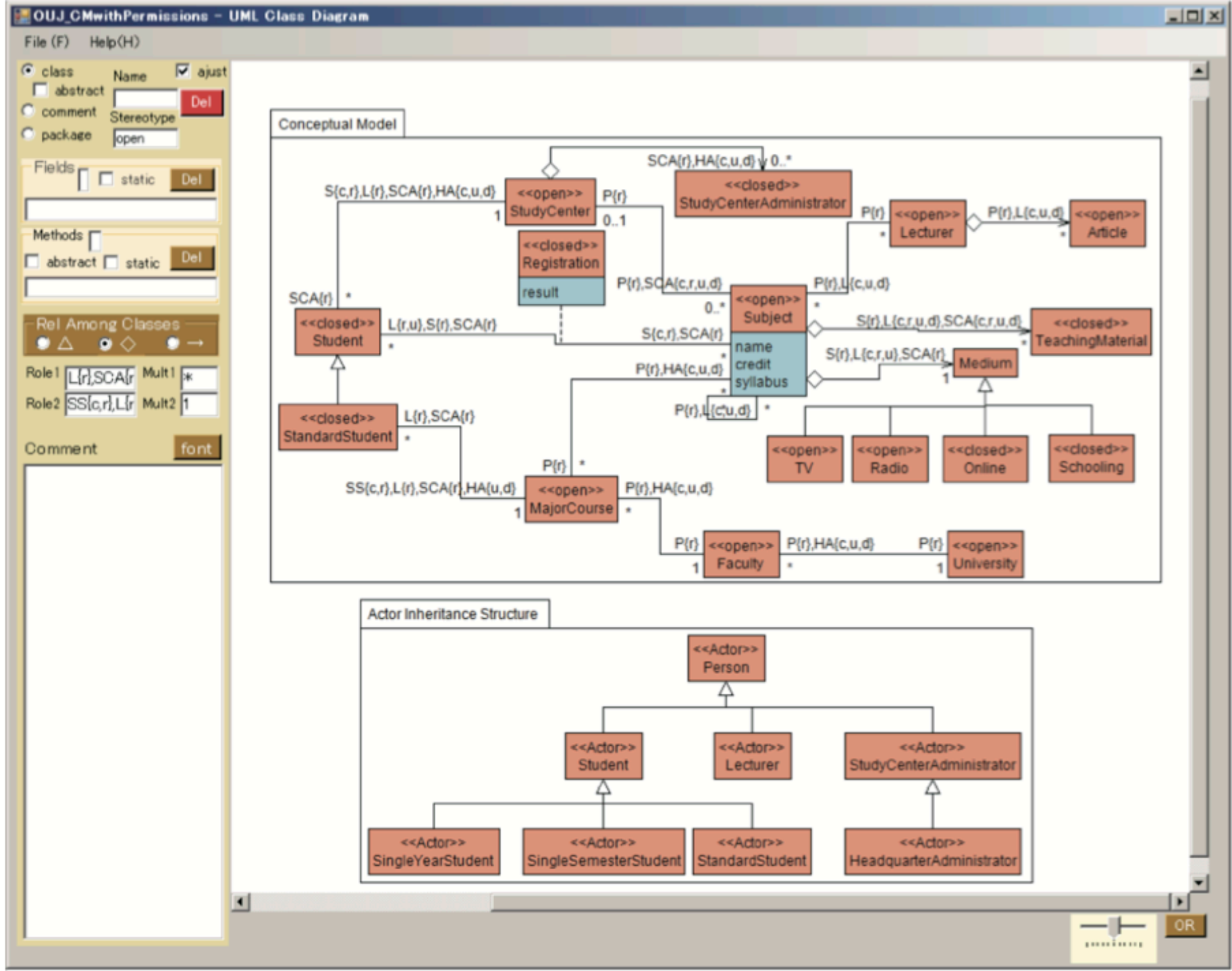
A process of the method

1. Develop a conceptual model agreed on by domain experts.
2. Define actors' groups in an inheritance structure.
   - The structure implies access permissions of actors.
3. Define permissions for every actors' category to access relations.
4. Traverse classes by tracing associations/aggregations via classes for each permitted actor and construct possible traverses for each actors' category.
5. Validate specified requirements with derived traverses.

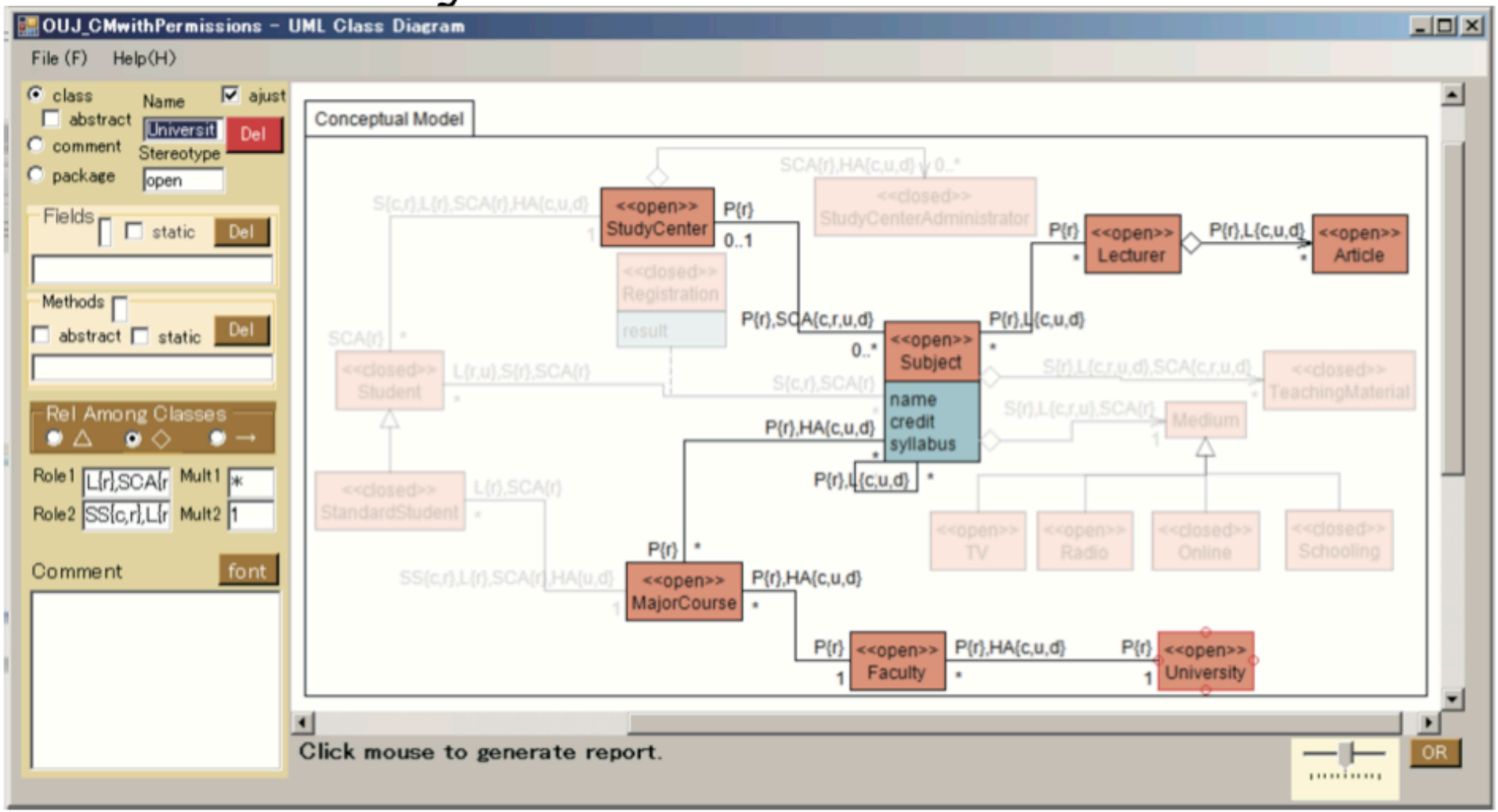
Experiment: System WAKABA for academic affairs of the Open University of Japan.

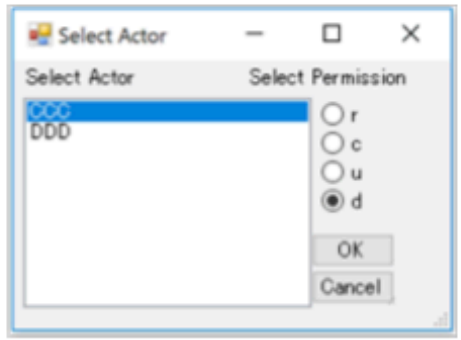
- It was newly released in March/2018.
- Users:
  - Students: approximately 90,000 (from 15 year-old to 100 or more? year-old)
  - Lectures, officers, and administrators: hundreds,
  - Expected visitors: All of Japanese 120,000,000
- We built a conceptual model with the scope of the system WAKABA.
  - Assumption: The model is correct, or at least adequate.

The conceptual model of OUJ



The visualized scope of Person's traverse from University





An example of a dialog to generate traverse paths. The paths are reported visually and also in a text file.

Unimplemented functional requirements

- MissReq1:
  - S/he cannot access the multimedia teaching materials from the syllabus.
- MissReq2:
  - S/he cannot access lecturers' information from the syllabus. Every syllabus is provided by a PDF file without any links.
- MissReq3:
  - In genera, a standard student who logs in to the system, s/he is not allowed to access open information.

Traverser could define.....

- An actor *StandardStudent* who accesses a/an *Subject* object can access(read) the *Medium* object or the *TV* object or the *Radio* object or the *Schooling* object or the *Online* object from the *Subject* object.
- An actor *Lecturer* who accesses a *Subject* object, can access the *Student* objects from the *Subject* object, and the *Registration* object of the *Subject* object for each *Student* object.
- An actor *Student* who accesses the *Student* object can access the *Subject* objects from the *Student* object, as well as the *Registration* object of each *Subject* object.
- An actor *Student* who accesses the *Student* object can access the *Subject* objects from the *Student* object, as well as the *Student* objects from each *Subject* object.

Discussion

- The threats of internal validity
  - The conceptual model of WAKABA might be built for traverse generation, only?
- The threats of external validity
  - Can every engineer use the method?
  - The undefined requirements that the Traverser generated were known to the stakeholders of WAKABA, but were deleted because of their priority policy.
  - If the size of the conceptual model becomes too large, the number of traverses will explode.
  - The undefined requirements in WAKABA are OUJ specific requirements.

Limitations of the method

- Since the conceptual model is a static model, we cannot generate traverses that imply temporal information. The tool can generate only possible traverses.
- The access control is imperfect, because the method only mentions permissions on associations/aggregations.

Future work

- Apply the method and tool to other domains and evaluate the effectiveness more widely.
- Validate the practicability of our approach with practitioners.

References

- Requirements Engineering literatures are available.
- Kenta Goto, Simpei Ogata, Junko Shirogane, Takako Nakatani, and Yoshiaki Fukazawa. 2015. Support of scenario creation by generating event lists from conceptual models. In 2015 3rd International Conference on Model-Driven Engineering and Software Development (MODELSWARD). Springer, 376–383.
- Takako Nakatani, Hideo Goto, Osamu Shigo, and Taichi Nakamura. 2018. Generating Scenarios with Access Permission from a Conceptual Model. In Proc. of the 12th Joint Conference on Knowledge-Based Software Engineering (JCKBSE2018), Maria Virvou, Fumihiro Kumeno, and Konstantinos Oikonomou (Eds.). Springer, 127–136.
- Takako Nakatani and Toshihiko Tsumaki. 2014. Predicting Requirements Changes by Focusing on the Social Relations. In Proc. of the 10th Asia-Pacific Conference on Conceptual Modeling. Australian Computer Society, 65–70.