TECHNISCHE UNIVERSITÄT DRESDEN

Software Technology Group

Faculty of Computer Science Institute of Software and Multimedia Technology, Software Technology Group

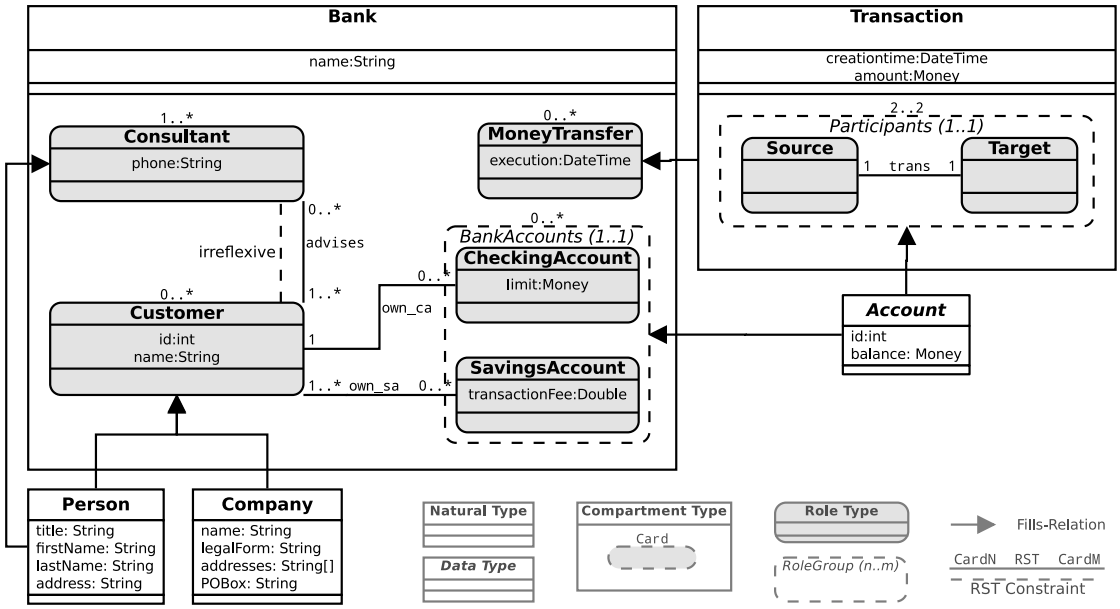
WS2019/20 – Design Patterns and Frameworks

Role-based Modeling of Design Patterns

Professor: Prof. Dr. Uwe Aßmann
Lectuer: Dr.-Ing. Sebastian Götz
Tutor: Dr. rer. nat. Marvin Triebel

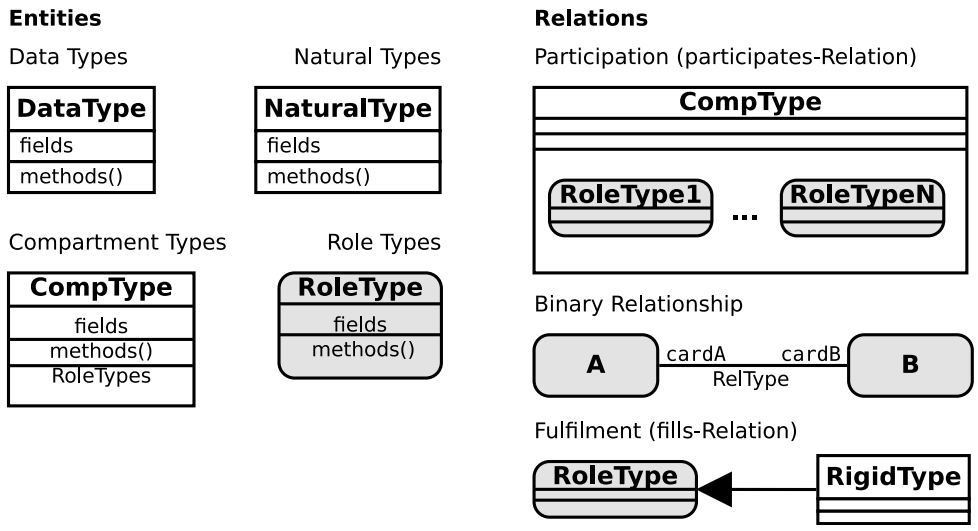
Task 1 Role-based Modeling

This exercise focuses on the *Compartment Role Object Model* (CROM) [2]. To illustrates CROM, the following figure showcases the role model of a banking application, extracted from [2, Fig. 2a].

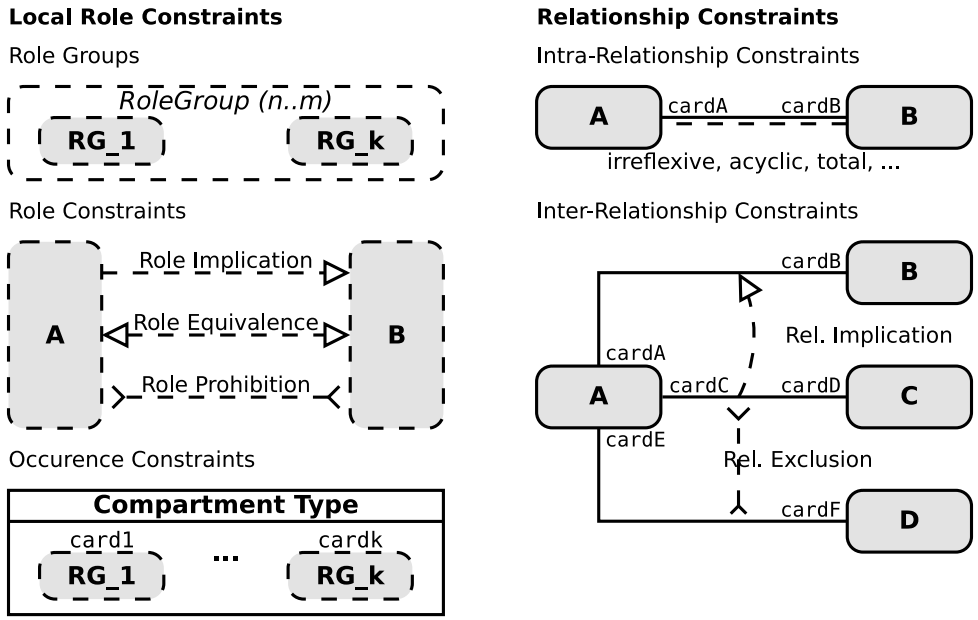


Furthermore, here is an overview of the notation of CROM highlighting all model elements, model relations, and model constraints.

The core entities and relations of CROM models are depicted below:



Additionally, CROM supports the following modeling constraints:



a) Read and understand Section 4 introducing CROM [2, Sect. 4].

Solution: Solved, if you have read Section 4 in [2].

Task 2 Role-based Horse Shows

To learn using the CROM language, the next task is to use roles to model *horse shows*.

In the world of horse shows, there are horses, persons, teams, and shows. A team encompasses exactly one *rider* (a person) and at least one *ridden* horse. Moreover, teams can enter a horse show as a *participant*, where they get a starting number . Furthermore, a horse show enrolls several *referees* (persons) to ensure that each team is examined by at least one referee. In fact, as the average horse show takes about two to three days, both horses and persons needed a place to stay. Consequently, the organization team is additionally tasked to provide *accommodations* for both horses and persons.

a) Classify the various concepts in the *horse show* domain in accordance to the ontological foundation [2, Sect. 4.1] as either compartment, role, or natural type.

Solution: Using the meta-properties *rigidity*, *foundedness*, and *identity* of the domain concepts [2, Sect. 4.1], each concept can be sorted into either of the following entity kinds.

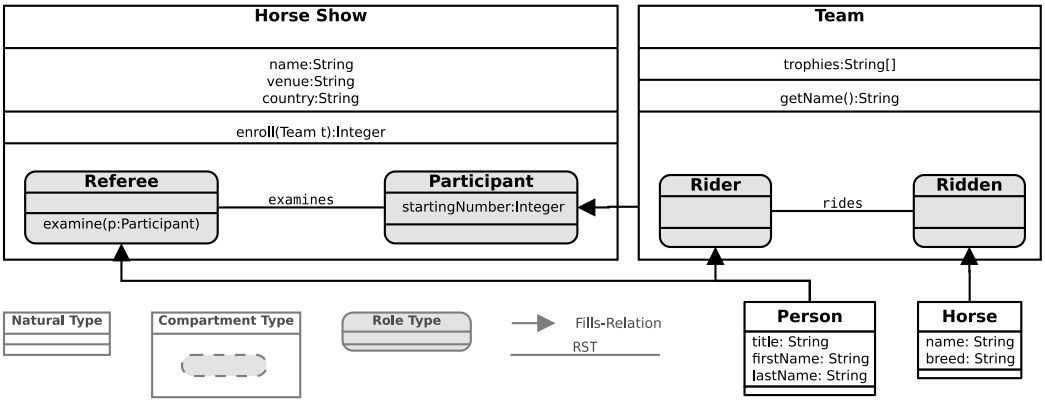
Table 1: Ontological classification of horse show domain concepts.

Concept	**Rigid**	**Founded**	**Identity**	**Domain Elements**
Natural Types	yes	no	unique	*person, horse, hotel, tent, horse stable*
Role Types	no[1]	yes	derived	*rider, ridden, participant, referee*
Compartment Types	yes	yes	unique	*team, show, accommodation*
Relationship Types	yes	yes	composed	*rides, examines*

[1] According to Guizzardi et al. role types are classified as anti-rigid [1].

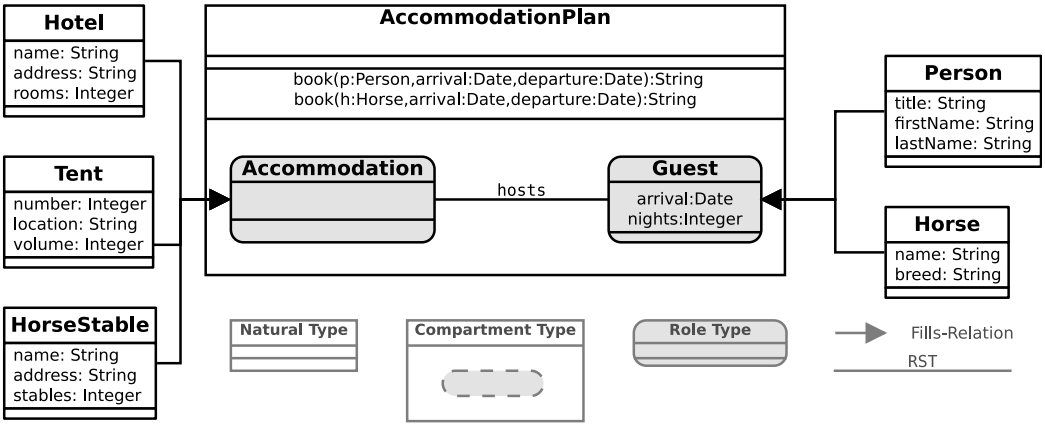
b) Design a role model for the *horse show* including persons, horses, teams, and shows.

Solution: The following CROM model represents the domain model for the horse show scenario.



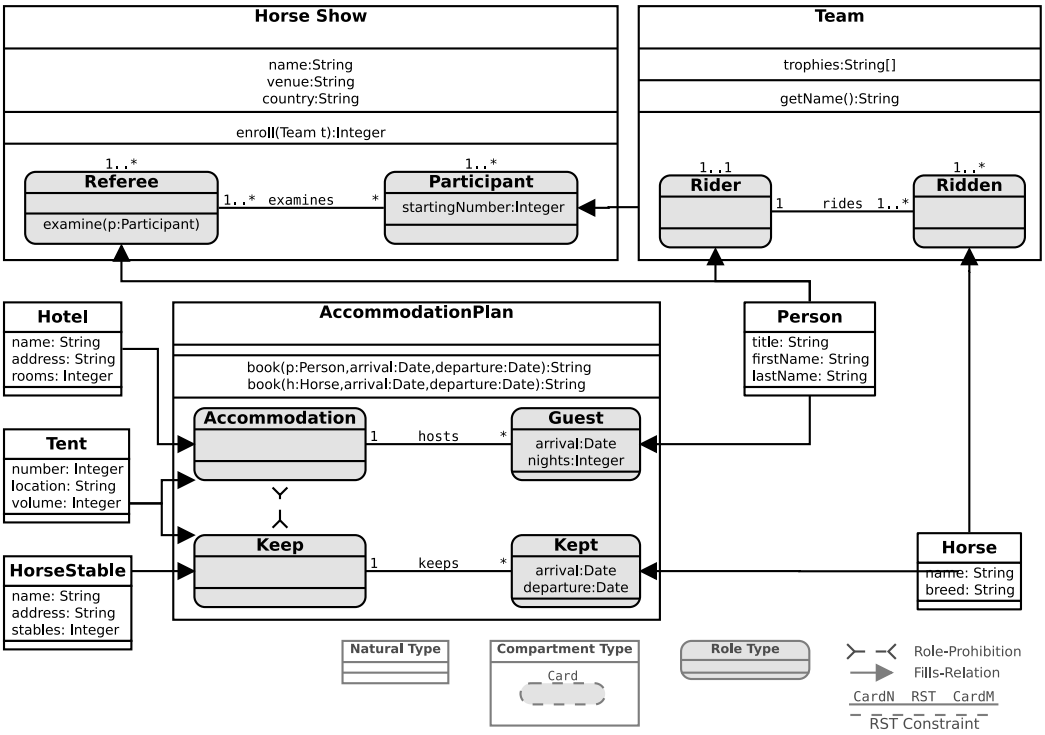
c) Extend the previous role model to additionally model accommodations for horses and persons, such as hotels, tents, and horse stables.

Solution: To extend the horse show CROM model, the Compartment Type *accommodation plan* is added where both persons and horses are *guests* accommodated by an *accommodation*, such as hotels, tents, or horse stables. The CROM model for managing accommodations is sketched as:



d) Finally, revise the role model and include the various constraints declared in the *horse show* domain.

Solution: Following the domain description, we first add *cardinalities* to relationships. Afterwards, *occurrence constraints* are defined within teams, horse shows and accommodation. Last but not least, the accommodation Compartment Type is extended to prohibit that horses are accommodated in hotel rooms and tents cannot accommodate both humans and horses. The completed horse show CROM model is presented below:



Task 3 Homework (optional)

The homework assigns you to further familiarize yourself with the CROM language [2].

a) As a formal modeling language, CROM supports the formal representation of role models. Using this formal notation, specify the role model designed in Task 2 by defining a corresponding CROM model $\mathcal{H}$ and constraint model $\mathcal{C}_{\mathcal{H}}$.

Solution: Let $\mathcal{H} = (NT, RT, CT, RST, \mathit{fills}, \mathit{parts}, \mathit{rel})$ be the CROM defined in Task 2; then the individual components are defined as follows:

$$
\begin{aligned}
NT &\coloneqq \{\text{Person}, \text{Horse}, \text{Hotel}, \text{Tent}, \text{HorseStable}\} \\
RT &\coloneqq \{\text{Referee}, \text{Participant}, \text{Rider}, \text{Ridden}, \text{Accommodation}, \text{Guest}, \text{Keep}, \text{Kept}\} \\
CT &= \{\text{Horse Show}, \text{Team}, \text{AccommodationPlan}\} \\
RST &= \{\text{examines}, \text{rides}, \text{hosts}, \text{keeps}\} \\
\mathit{fills} &\coloneqq \{(\text{Person}, \text{Referee}), (\text{Team}, \text{Participant}), (\text{Person}, \text{Rider}), \dots\} \\
\mathit{parts} &\coloneqq \{\text{Horse Show} \to \{\text{Referee}, \text{Participant}\}, \text{Team} \to \{\text{Rider}, \text{Ridden}\}, \dots\} \\
\mathit{rel} &\coloneqq \{\text{examines} \to (\text{Referee}, \text{Participant}), \text{rides} \to (\text{Rider}, \text{Ridden}), \\
&\qquad \text{hosts} \to (\text{Accommodation}, \text{Guest}), \text{keeps} \to (\text{Keep}, \text{Kept})\}
\end{aligned}
$$

Afterwards, $\mathcal{C}_{\mathcal{H}} = (\mathit{rolec}, \mathit{card}, \mathit{intra})$ is the constraint model of the horse show CROM, where the components are defined as:

$$
\begin{aligned}
\mathit{rolec} &\coloneqq \{\text{Horse Show} \to \{(1..\infty, \text{Referee}), (1..\infty, \text{Participant})\}, \\
&\qquad \text{Team} \to \{(1..1, \text{Rider}), (1..\infty, \text{Ridden})\}, \\
&\qquad \text{AccommodationPlan} \to \{(0..\infty, (\{\text{Accommodation}, \text{Keep}\}, 1, 1)\}\} \\
\mathit{card} &\coloneqq \{\text{examines} \to (1..\infty, 0..\infty), \text{rides} \to (1..1, 1..\infty), \\
&\qquad \text{hosts} \to (1..1, 0..\infty), \text{keeps} \to (1..1, 0..\infty)\} \\
\mathit{intra} &\coloneqq \emptyset
\end{aligned}
$$

References

[1] Giancarlo Guizzardi. *Ontological foundations for structural conceptual models.* CTIT, Centre for Telematics and Information Technology, 2005.

[2] Thomas Kühn, Böhme Stephan, Sebastian Götz, and Uwe Aßmann. A combined formal model for relational context-dependent roles. In *Proceedings of the 2015 ACM SIGPLAN International Conference on Software Language Engineering*, pages 113–124. ACM, 2015. doi: 10.1145/2814251.2814255. URL http://dl.acm.org/citation.cfm?id=2814255.

[3] Dirk Riehle and Thomas Gross. Role model based framework design and integration. In *ACM SIGPLAN Notices*, volume 33, pages 117–133. ACM, 1998.