Knowledge Engineering (SCOM7348)

Final Check & Schema Engineering Issues

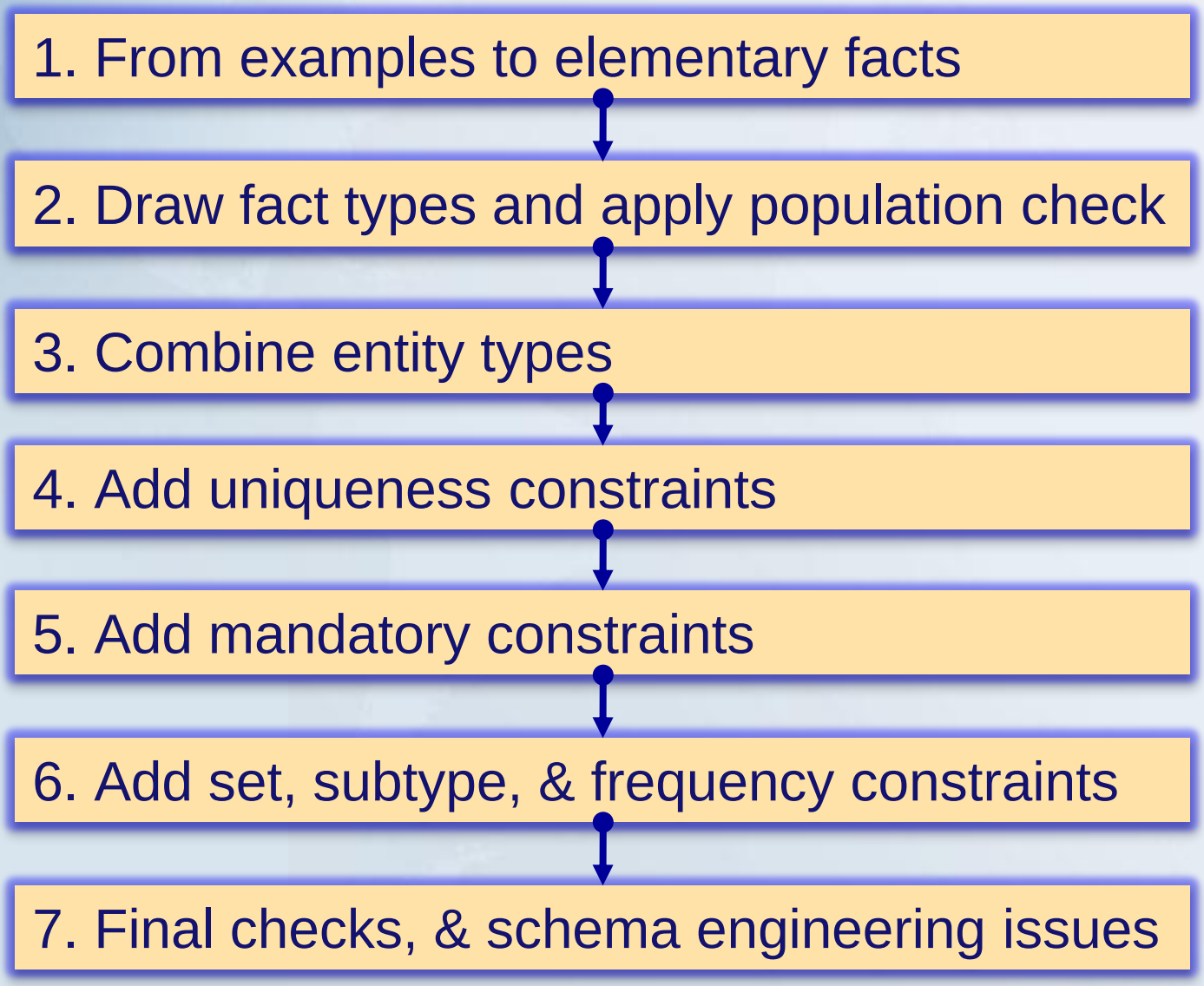
Chapter 7, Chapter 12, & Selected Topics

Dr. Mustafa Jarrar

University of Birzeit

mjarrar@birzeit.edu
www.jarrar.info

Conceptual Schema Design Steps



Outline

- Final Check
  - Rules Implications
  - Rules Contradictions
  - Modeling Tips (Check List)
- Rules Verbalization
- Schema equivalence and Optimization
- Schema Modularization

Final Checks

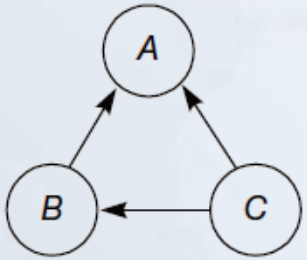
- No constraint contradict the other.

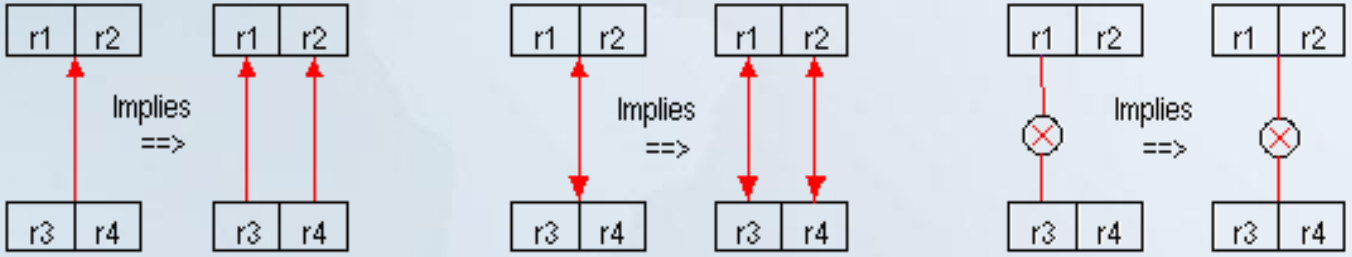
  No constraint implies the other.

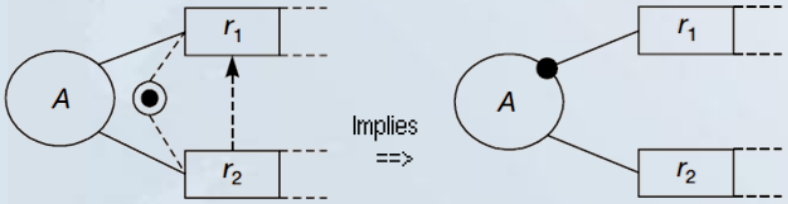
  Other Modeling Tips (Check List)

Constraint Implications (Examples)

Some constraints may imply each other (see cases below), the implied constraint should be removed because it complicates the model without bringing no value.







➔ Many different examples are given in previous chapters

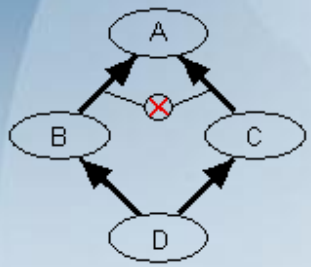
Outline

- Final Check
  - Rules Implications
  - Rules Contradictions
  - Modeling Tips (Check List)
- Rules Verbalization
- Schema Equivalence and Optimization
- Schema Modularization

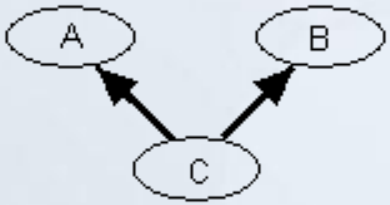
Constraint Contradictions (Examples)

Based on [1]

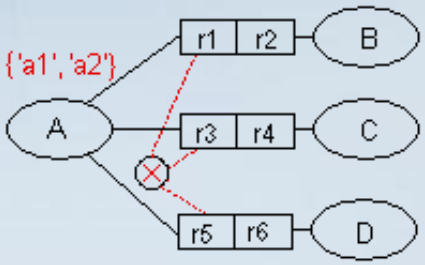
Some constraints may contradict each other (see cases below).



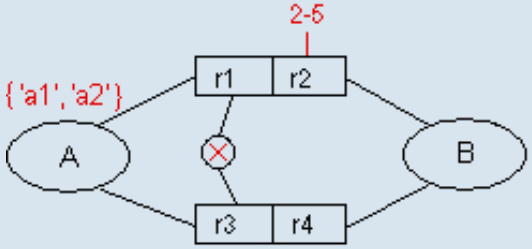
D will never be populated, because of the exclusive constraint



C will never be populated, because A ad B ad disjoint by definition.



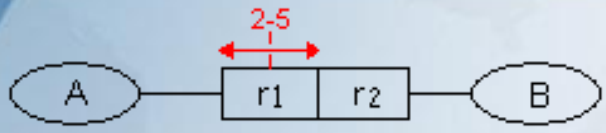
One of the roles (r1, r2, or r3) will never be populated, because we have only two values possible ‘a1’ and ‘a2’



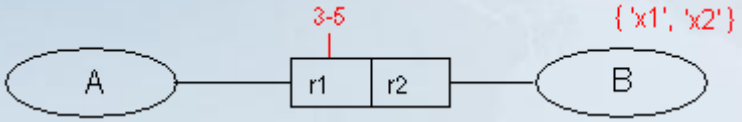
Due to the frequency constraint, there should be at least two different values to populate r1. In order to populate r3, we need, by the exclusion constraint, a value different from the two for role r1. In total, we thus need three different values in order to be able to populate both r1 and r2, but this contradicts with the value constraint on object-type A: we only have 2 values at our disposal.

Constraint Contradictions (Examples)

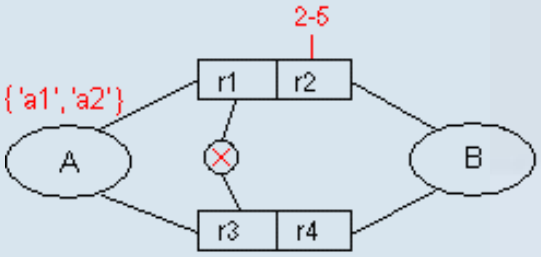
Based on [1]



the uniqueness constraint indicates that the role r1 should be played by at most one element, while the frequency constraint demands that there are at least 2 and at most 5 participants in the role. It is thus impossible to populate r1.



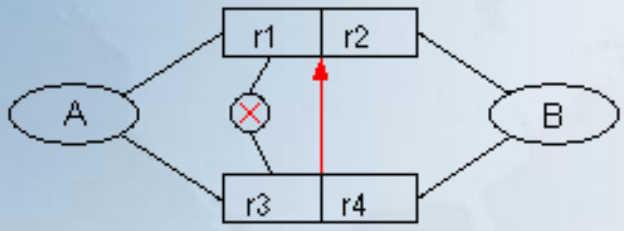
If the frequency constraint 3-5 on r1 is satisfied, each instance of A must play r1 at least three times, and thus three different instances of B are required. However, there are only two possible instances of B, which are declared by the value constraint {‘x1’, ‘x2’}. Thus r1 cannot be populated.



Who can tell where the contradictions?

Constraint Contradictions (Examples)

Based on [1]



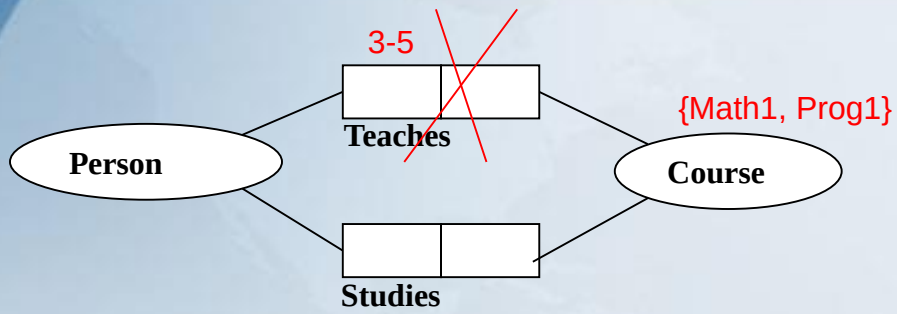
The exclusion constraint between the two roles r1 and r3 means that their populations should be distinct. However, in order to satisfy the subset constraint between the relations (r1; r2) and (r3; r4), the populations of r1 and r3 should not be distinct. In other words, the exclusion constraint between roles r1 and r3 implies an exclusion constraint between the relations (r1; r2) and (r3; r4), which contradicts any subset or equality constraint between both predicates.

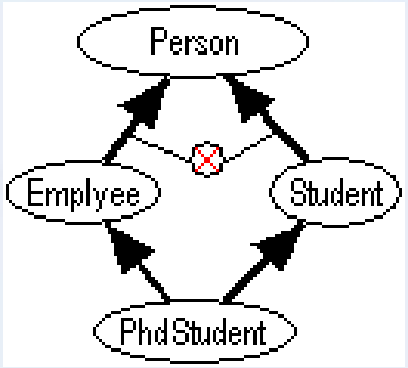
➔ Many different examples are given in previous chapters

➔ Any Idea to detect such contradictions automatically?

Reasoning on ORM Schemes

Based on [2]





- Schema satisfiability: A schema is satisfiable if and only if there is at least one concept in the schema that can be populated. ➔ Weak satisfiability
- Concept satisfiability: A schema is satisfiable if and only if all concepts in the schema can be populated.
- Role satisfiability: A schema is satisfiable if and only if all roles in the schema can be populated. ➔ Strong satisfiability

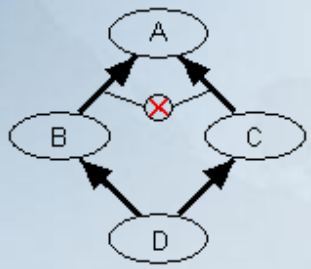
➔ Concept satisfiability implies schema **satisfiability** .

➔ Role satisfiability implies concept **satisfiability** .

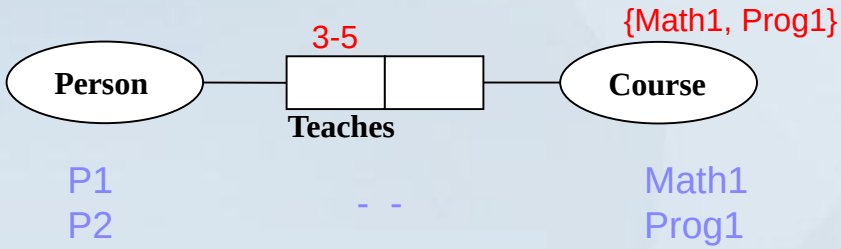
Schema Satisfiability

➔ Weak satisfiability

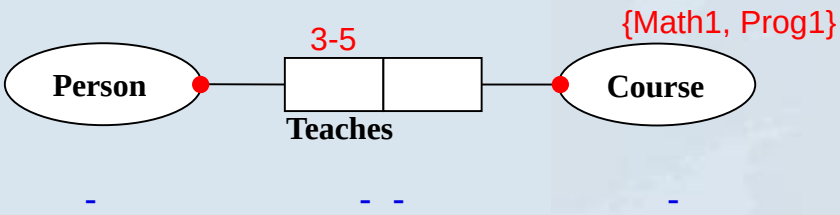
A schema is satisfiable if and only if there is at least one concept in the schema that can be populated.



✓ **Schema-Satisfiable,**
because A, B, and C can be populated



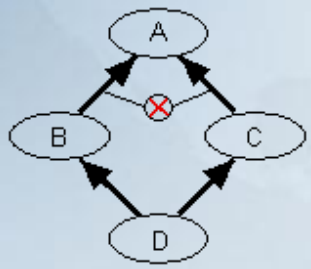
✓ **Schema-Satisfiable,**
As both concepts alone (Person & Courses) can be populated, although the roles cannot be populated.



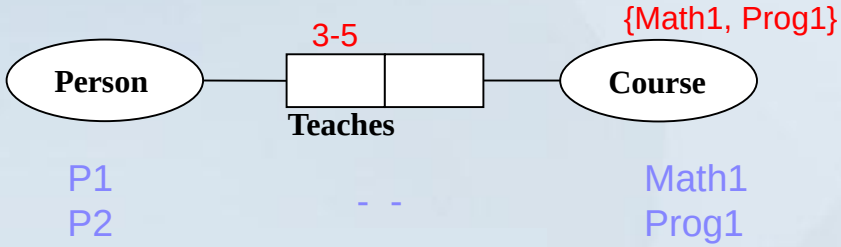
✗ **Schema-Unsatisfiable,**
As both concepts alone (Person &Courses) can be populated, although the roles cannot be populated.

Concept Satisfiability

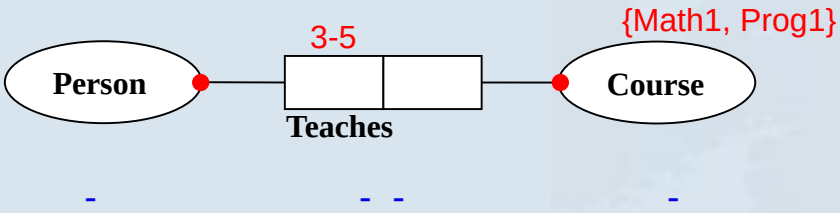
A schema is satisfiable if and only if all concepts in the schema can be populated.



✗ **Concept-Unsatisfiable,**
because there is one concept (i.e. D) that cannot be populated.



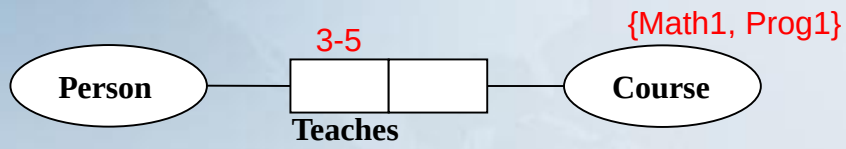
✓ **Concept-Satisfiable,**
As all concepts can be populated, although the roles cannot be populated.



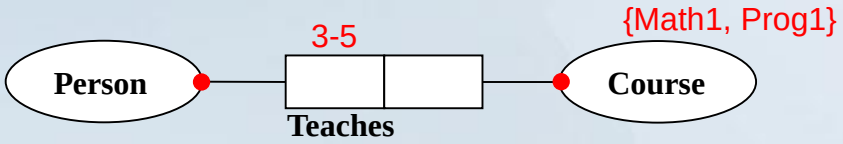
✗ **Concept-Unsatisfiable,**
As no concepts can be populated, because of the mandatory constraints.

Role Satisfiability

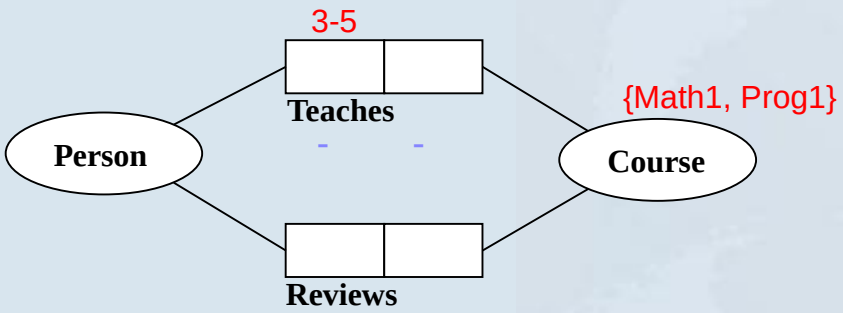
A schema is satisfiable if and only if all roles in the schema can be populated.



✗ **Role-Unsatisfiable,**
As no roles can be populated.



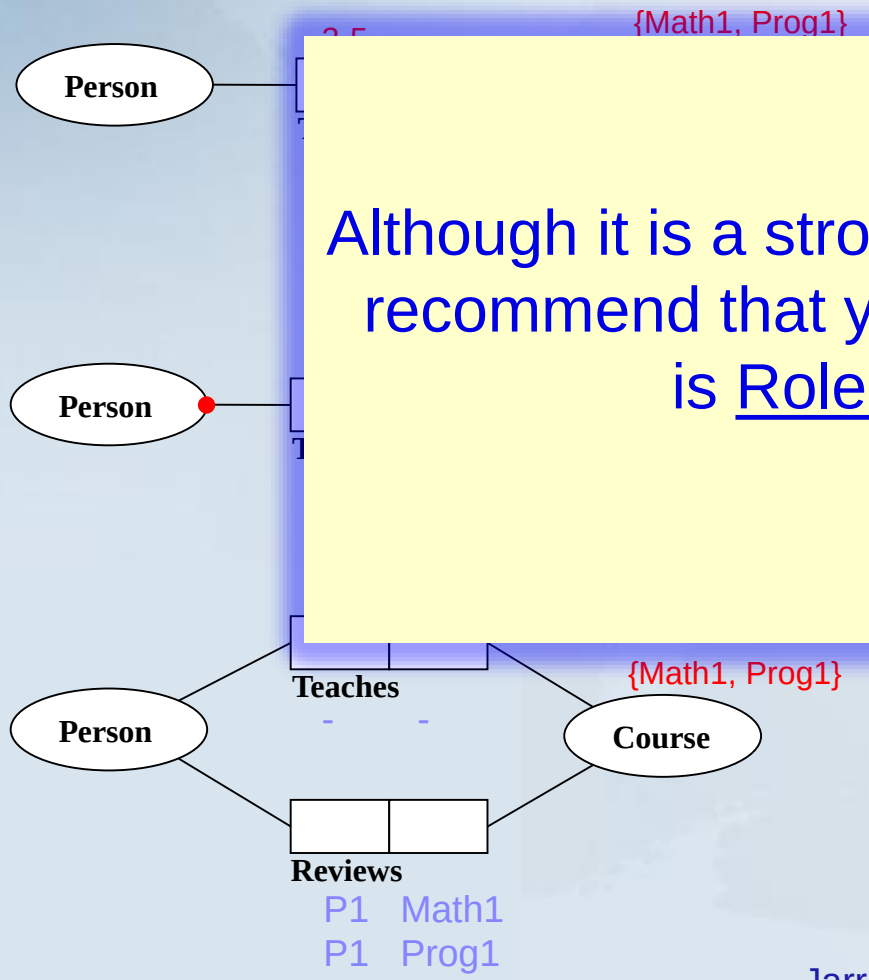
✗ **role-Unsatisfiable,**
As all roles cannot be populated.



✗ **role-Unsatisfiable,**
As not all roles can be populated.

Role Satisfiability

A schema is satisfiable if and only if all roles in the schema can be populated.



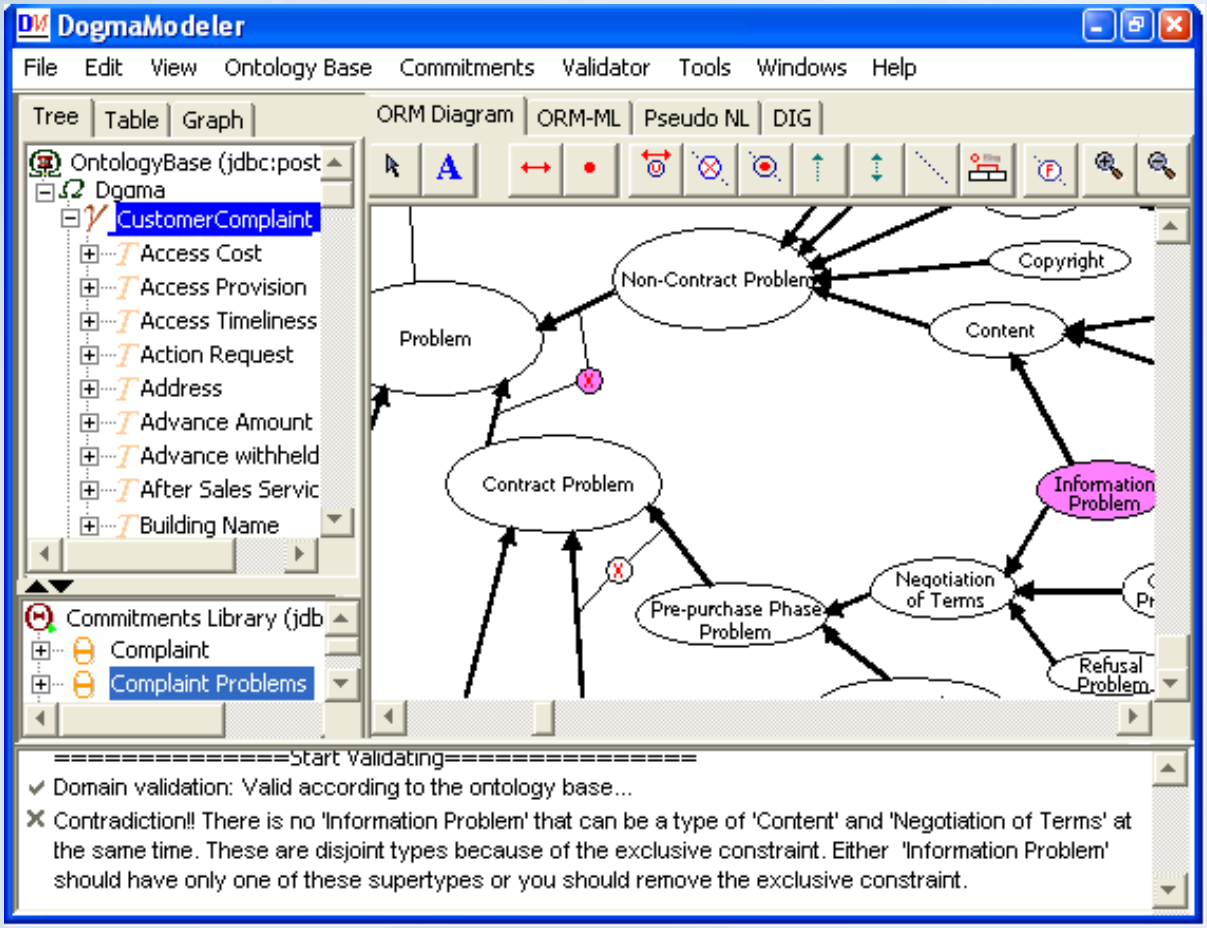
Although it is a strong requirement, but we recommend that your conceptual model is Role Satisfiable.

As some roles can be populated.

DogmaModeler

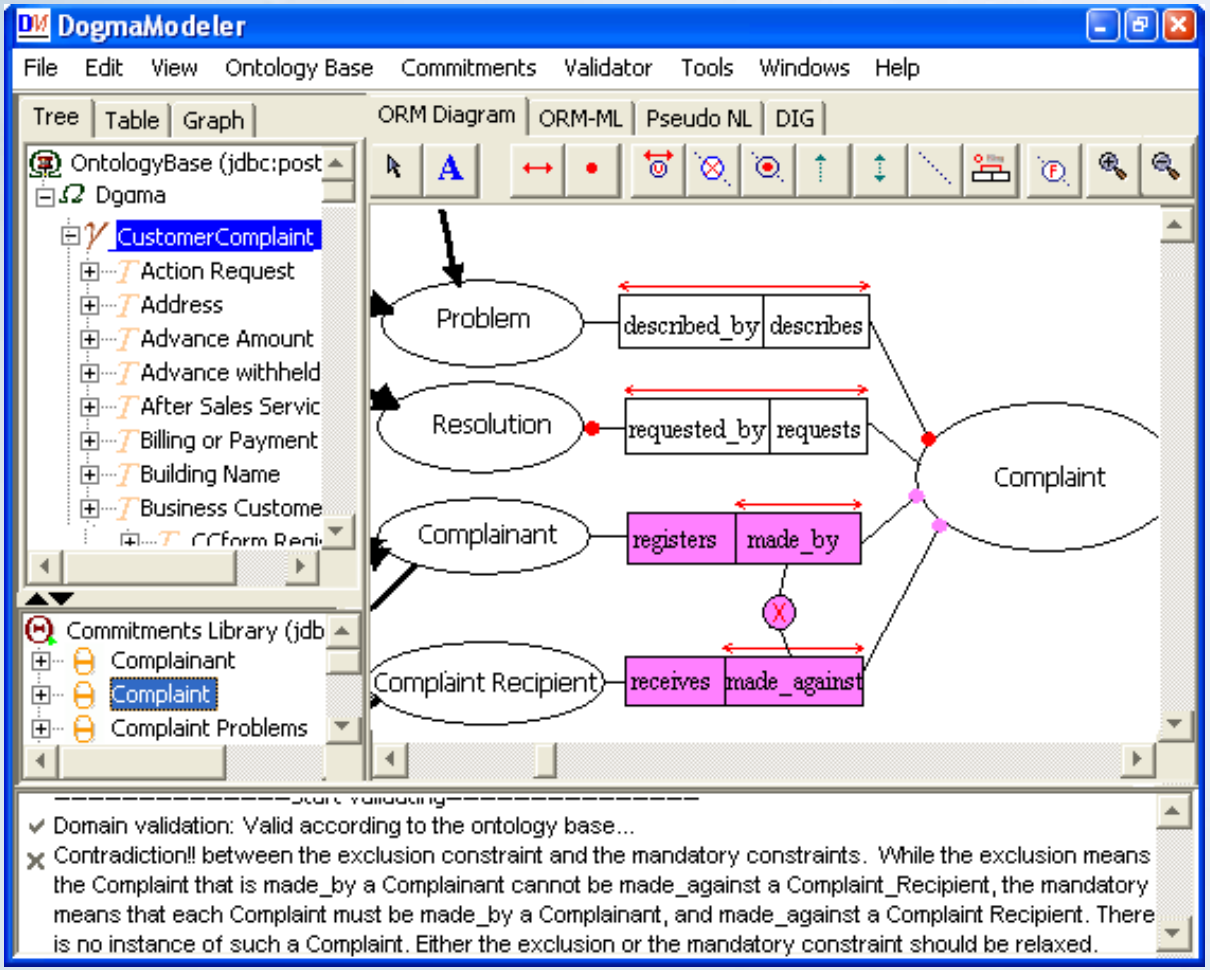
http://www.jarrar.info/Dogmamodeler/

Is the only tool that can detect constraint contradiction for ORM



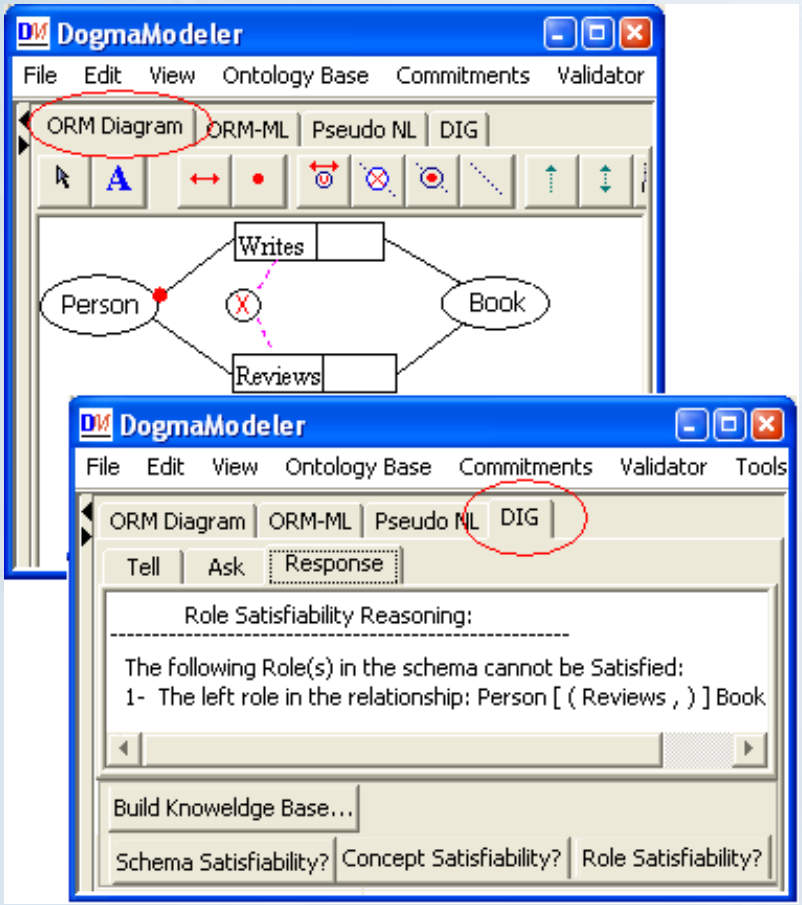
DogmaModeler

Is the only tool that can detect constraint contradiction for ORM



DogmaModeler

Is the only tool that can detect constraint contradiction for ORM



Outline

- Final Check
  - Rules Implications
  - Rules Contradictions
  - Modeling Tips (Check List)
- Rules Verbalization
- Schema Equivalence and Optimization
- Schema Modularization

Modeling Tips and Common Mistakes
(for beginners)

Based on [3]

- Check each role in the model, whether it should be unique?
- Check each role in the model, whether it should be Mandatory?
- Check each entity (Object Type) whether it has an identity?
- Check each leaf nodes whether should be Value Type?
- Check each value constraint whether it placed on Value Type only?
- The syntax of values and ranges in value constraints is correct.
- Check each subtype, that it is playing some roles.
- External uniqueness and disjunctive mandatory constraints are placed on the correct roles.
- Preferred: If you have subtypes, then their supper type should have a value constraint.

Modeling Tips and Common Mistakes (for beginners)

Based on [3]

- Role names:
  - At least one role, in each relation, has a label.
  - Names should be correct, expressive, and meaningful
  - Naming style: for example “WorksFor”, “AffiliatedWith”, “IsOf”, etc.
- Concept Names:
  - Should be expressive and meaningful (as used in the domain), correct translation
  - Naming style: for example “FacultyMember”, “NaturalPerson”
  - Don’t use plural as concept labels (e.g., students, courses)
- Readability\Beauty of the Diagrams
  - place related properties beside each other (country, city…) or (name, fname, lname).
  - Flip roles if needed.
  - Lines are straight, and the whole diagram is balanced (as much as you can)
  - Page layout is landscape if needed.
  - The sizes of the concepts are equal, unless you what to emphasize the main concepts.
  - Important concepts are placed in the middle, and all concepts are aligned.
  - Roles are aligned and similar roles have the size.
  - Populate a page as much as you can (BUT NOT too much)
  - Do not clone concepts if not necessary
  - Modularize a large diagram into pages (but keep very related concepts in the same page).first pages contain the most important
  - Write your project details (name, course, year, project#, date,….) in each page.

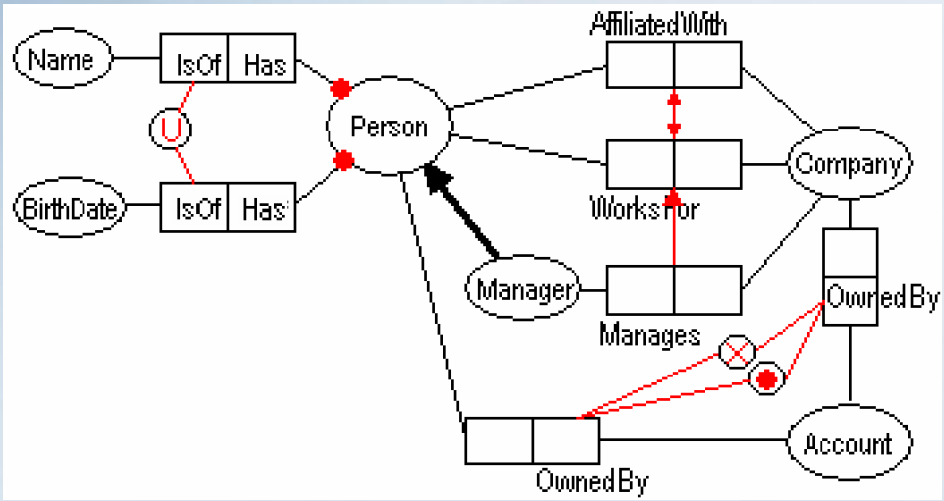
Outline

- Final Check
  - Rules Implications
  - Rules Contradictions
  - Modeling Tips (Check List)
- Rules Verbalization
- Schema Equivalence and Optimization
- Schema Modularization

Rules Verbalization

Based on [3]

Verbalization is the process of writing the semantics captured by the ORM constrains as pseudo-natural language (fixed-syntax) sentences.



Notice that these verbalizations can be generated atomically using fixed templates

- *Subsumption*: **Each** Manager **must be a type of** Person.
- *Mandatory*: **Each** Person **must** Has **at least one** Name.
- *Mandatory*: **Each** Person **must** Has **at least one** BirthDate.
- *InterUniqueness*: **The combination of {**BirthDate, Name**} must refer to at most one** Person.
- *Equality*: **Each** Person WorksFor **a** Company **must** AffliatedWith **that** Company, **and vice versa.**
- *Subset*: **Each** Manager **who** Manages **a** Company **must** WorksFor **that** Company.
- *ExMandatory*: **Each** Account OwnedBy Person **or** OwnedBy Company**, or both**.
- *Exclusion*: **No** Account **can be** OwnedBy **a** Company **and** OwnedBy **a** Person.

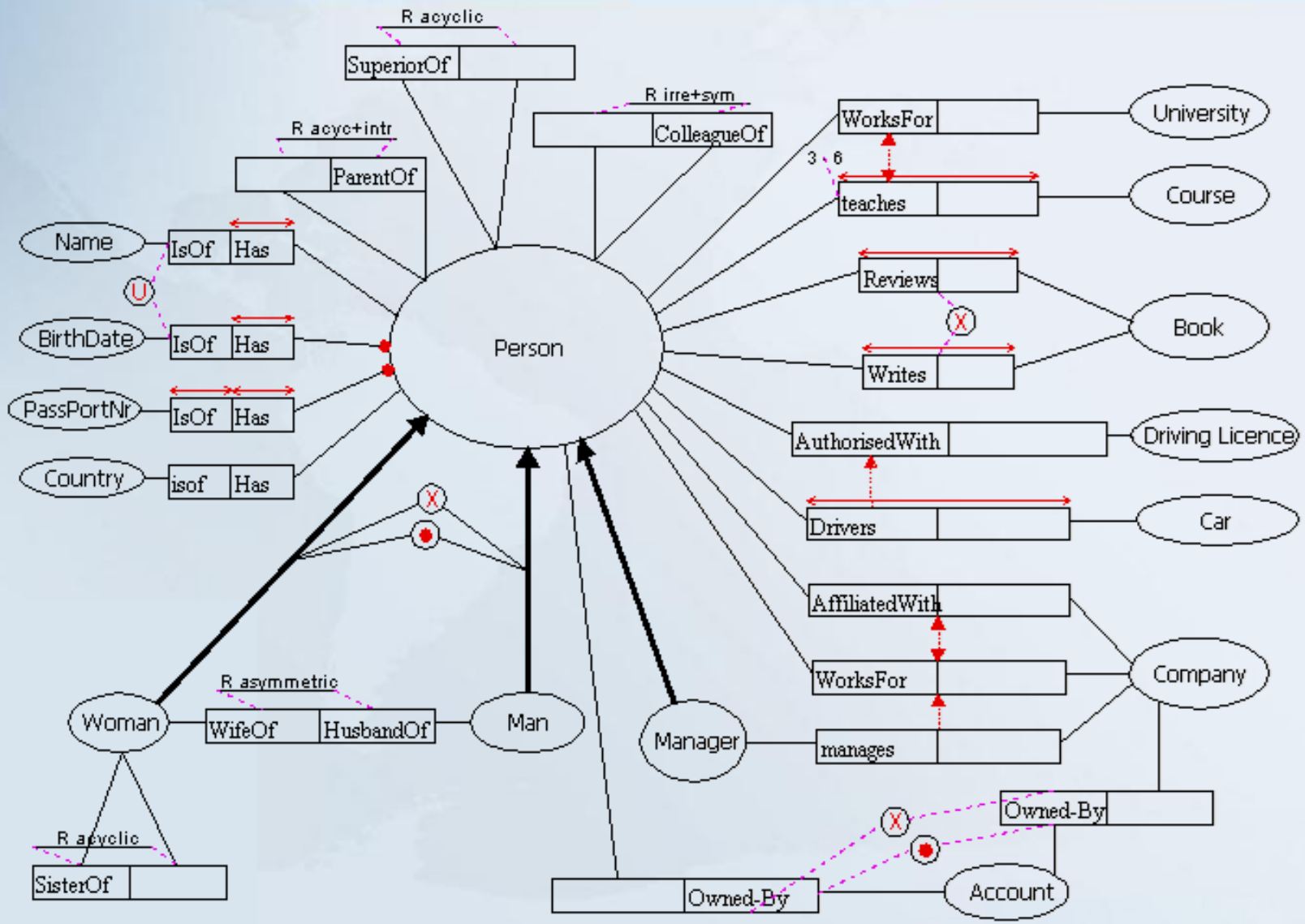
Rules Verbalization

Verbalization is the process of writing the semantics captured by the ORM constrains as pseudo-natural language (fixed-syntax) sentences.

- This pseudo-natural language is understandable for domain experts, which enables them to help in the modeling process, as they can review whether the rules are correct.
- See **http://www.jarrar.info/orm/verbalization/** which offers templates for verbalizing ORM in 10 languages

- *Subsu*
- *Mand*
- *Mand*
- *InterUniqueness*: **The combination of {**BirthDate, Name**} must refer to at most one** Person.
- *Equality*: **Each** Person WorksFor **a** Company **must** AffliatedWith **that** Company, **and vice versa.**
- *Subset*: **Each** Manager **who** Manages **a** Company **must** WorksFor **that** Company.
- *ExMandatory*: **Each** Account OwnedBy Person **or** OwnedBy Company**, or both**.
- *Exclusion*: **No** Account **can be** OwnedBy **a** Company **and** OwnedBy **a** Person.

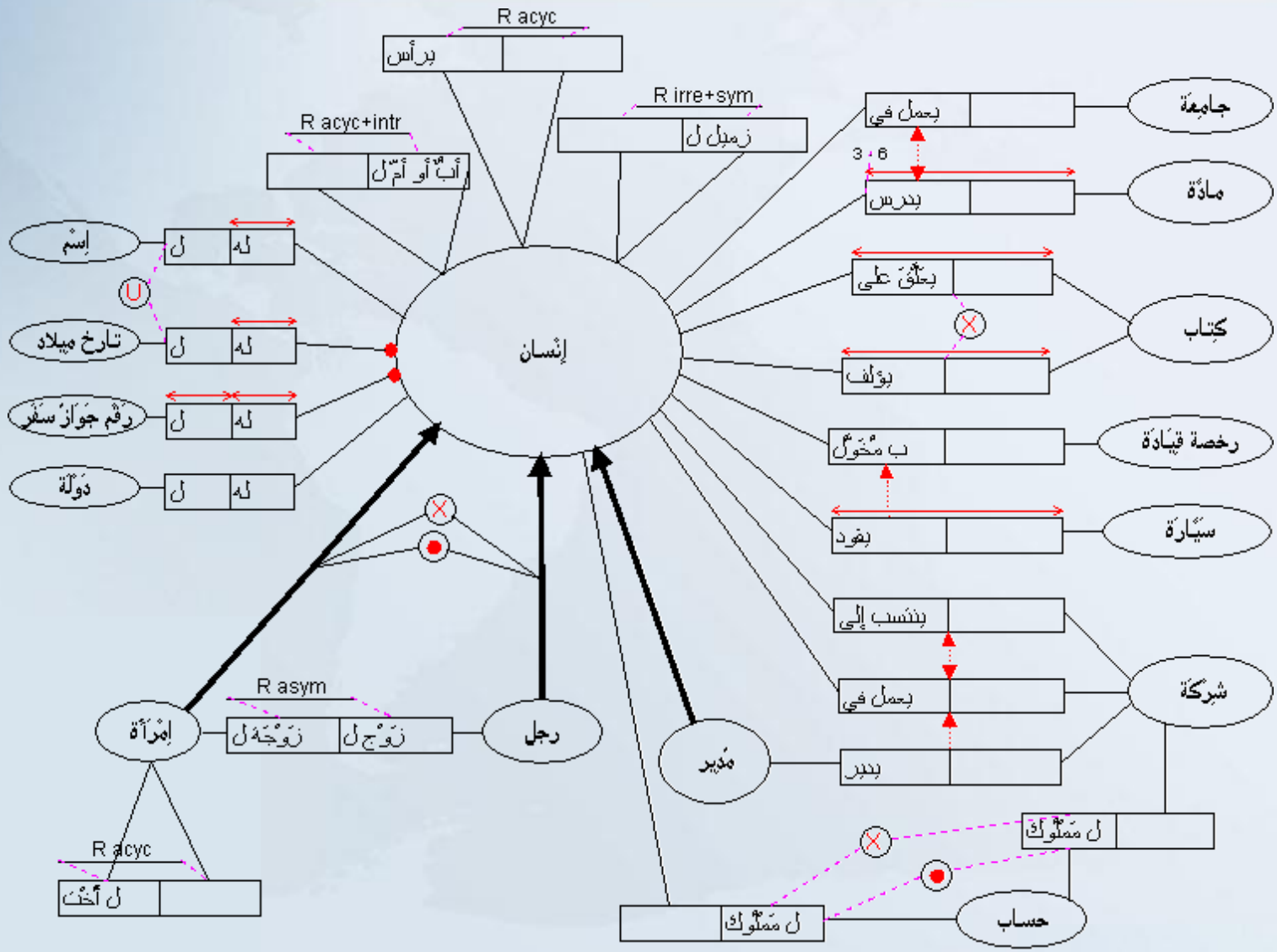
An ORM model with many constraints



Verbalization of the constraints (English)

-[Mandatory] Each Person must Has at least one PassPortNr.
-[Mandatory] Each Person must Has at least one BirthDate.
-[Mandatory] Each Account should be Owned-By Company or Owned-By Person.
-[Uniqueness] Each Person must Has at most one BirthDate.
-[Uniqueness] Each Person must Has at most one Name.
-[Uniqueness] Each Person must Has at most one PassPortNr.
-[Uniqueness] Each PassPortNr must IsOf at most one Person.
-[Uniqueness] It is possible that Person teaches more than one Course , and vice versa.
-[Uniqueness] It is possible that Person Reviews more than one Book , and vice versa.
-[Uniqueness] It is possible that Person Writes more than one Book , and vice versa.
-[Uniqueness] It is possible that Person Drivers more than one Car , and vice versa.
-[Uniqueness] The combination of { BirthDate and Name } must refer to at most one Person.
-[Exclusive] Each Person should be either Woman or Man.
-[Totality] Each Person must be, at least, Man or Woman.
-[Subset] If Person Drivers Car then this Person AuthorisedWith Driving Licence.
-[Subset] If Manager manages Company then this Person WorksFor that Company.
-[Equality] Person WorksFor University if and only if this Person teaches Course.
-[Equality] Person AffiliatedWith Company if and only if this Person WorksFor that Company.
-[Exclusion] No Account Owned-By Company and also Owned-By Person.
-[Exclusion] No Person Writes Book and also Reviews that Book.
-[Value] The possible instances of Country are :{Belgium, France, Germany}
-[Irreflexive] No Person ColleagueOf it/him self.
-[Symmetric] If Person X ColleagueOf Person Y, it must be vice versa.
-[Acyclic] Person cannot be directly (or indirectly through a chain) SuperiorOf it/him self .
-[Acyclic] Woman cannot be directly (or indirectly through a chain) SisterOf it/him self .
-[Asymmetric] If Person X WifeOf Person Y, it cannot be vice versa .
-[Intransitive] If Person X ParentOf Person Y, and Y ParentOf Z, then it cannot be that X ParentOf Z.
-[Frequency] If Person Teaches Course, then this Person Teaches at least 3 and at most 6 Course(s).

Same Example in Arabic



Verbalization of all constraints (Arabic)

-**[Mandatory]** كل إنْسان له رَقْم جَوَازُ سَفَر الاقل
-**[Mandatory]** كل إنْسان له تارخ مِيلاد واحد على الاقل
-**[Mandatory]** كل حساب يجب ان يكون مملوك لانسان او مملوك لشركة
-**[Uniqueness]** كل انسان له تاريخ ميلاد واحد على الاكثر
-**[Uniqueness]** كل انسان له اسم واحد على الاكثر
-**[Uniqueness]** كل انسان له رقم جواز سفر واحد على الاكثر
-**[Uniqueness]** كل رقم جواز سفر لانسان واحد على الاكثر
-**[Uniqueness]** كل انسان يمكن ان يدرس اكثر من مادة والعكس صحيح
-**[Uniqueness]** كل انسان يمكن ان يؤلف اكثر من كتاب والعكس صحيح
-**[Uniqueness]** كل انسان يمكن ان يعلق على اكثر من كتاب والعكس صحيح
-**[Uniqueness]** كل انسان يمكن ان يقود اكثر من سيارة والعكس صحيح
-**[Uniqueness]** اتحاد كل من تاريخ ميلاد واسم يشير الى انسان واحد على الاكثر
-**[Exclusive]** كل انسان يمكن ان يكون اما رجل او اِمْرَأَة
-**[Totality]** كل انسان يجب ان يكون رجل او اِمْرَأَة
-[Subset] اذا انسان يقود سيارة فان هذا الانسان مخول برخصة سياقة
-**[Subset]** اذا مديريديرشركة فان هذا المديريعمل في هذة الشركة
-**[Equality]** كل انسان يعمل في جامعة اذا و فقط اذا هذا الانسان يدرس مادة
-**[Equality]** كل انسان منسوب لشركة اذا و فقط اذا هذا الانسان يعمل في هذة الشركة
-[Exclusion] لا يمكن ان يكون حساب مملوك لا نسان و في نفس الوقت مملوك لشركة
-**[Exclusion]** لا يمكن ان يكون انسان يعلق على كتاب و في نفس الوقت يؤلف ذلك كتاب
-**[Value]** القيم الممكنة ل دولة هي:**{** بلجيكا, فرنسا, المانيا**}**
-**[Irreflexive]** لا يجوز لانسان ان يكون زميل لنفسه
-**[Symmetric]** اذا انسان س زميل ل ص,فان العكس بالعكس
-**[Acyclic]** لايمكن لانسان ان يكون (بطريقة مباشرة او غير مباشرة) اب او ام لنفسه
-**[Acyclic]** لايمكن لانسان ان يكون (بطريقة مباشرة او غير مباشرة) مشرف على نفسه
-**[Asymmetric]** اذا انسان س زوجة لانسان ص, فان العكس غير صحيح
-**[Intransitve]** ل ج اذا انسان س اب او ام لانسان ص, و ص اب او ام لانسان ج, فانه لايمكن ان يكون س اب او ام
-**[Frequency]** اذا الانسان يدرس مادة, فان هذا الانسان يجب ان يدرس بين 2 الى 3 مادة

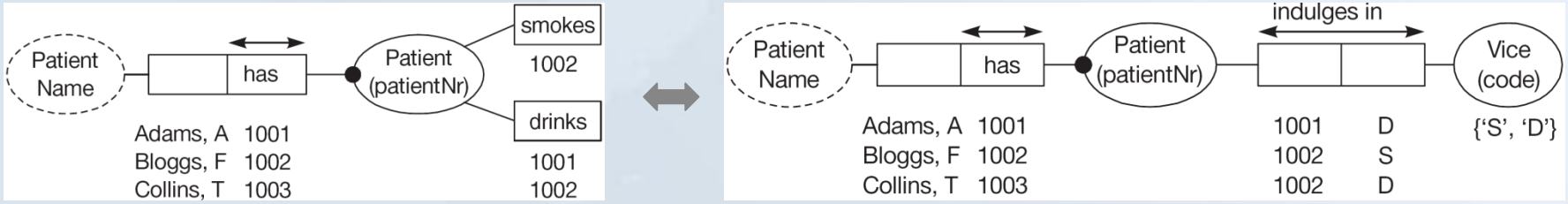
Outline

- Final Check
  - Rules Implications
  - Rules Contradictions
  - Modeling Tips (Check List)
- Rules Verbalization
- Schema Equivalence and Optimization
- Schema Modularization

Schema Equivalence and Optimization

Based on [2]

- It is not surprising that people often come up with different ways (i.e., deferent conceptual models) of describing the same reality.
- Two conceptual schemas are equivalent if and only if whatever UoD state or transition can be modeled in one can also be modeled in the other.
- What is the difference between these two schemes:



➢ The act of reshaping two equivalent schemes like this is said to be a **conceptual schema transformation.**

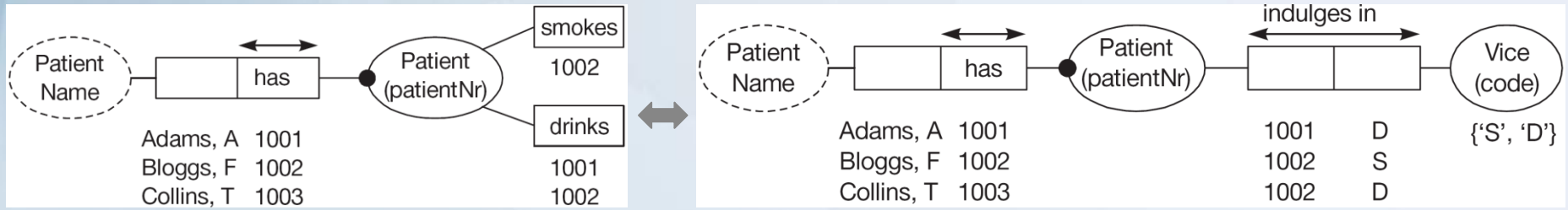
Schema Equivalence and Optimization

- Skills of schema transformations helps us to see what different design choices are possible.
- Moreover, if two independently developed schemas are to be either fully or partly integrated, we often need to resolve the differences in the ways that each schema models common UoD features.
- To do this, we need to know whether one representation can be transformed into the other, and if so, how.
- Another use of conceptual schema transformations is to reshape the original conceptual schema into one that maps directly to a more efficient implementation, or to more conceptually elegant schema.
- This process is known as **conceptual schema optimization**.

➔There are two class of schema transformations:
Predicate Specialization, and **Predicate Generalization**

Predicate Specialization and Generalization

If two or more predicates may be thought of as special cases of a more general predicate, then we may replace them by the more general predicate, so long as the original distinction can be preserved in some way.

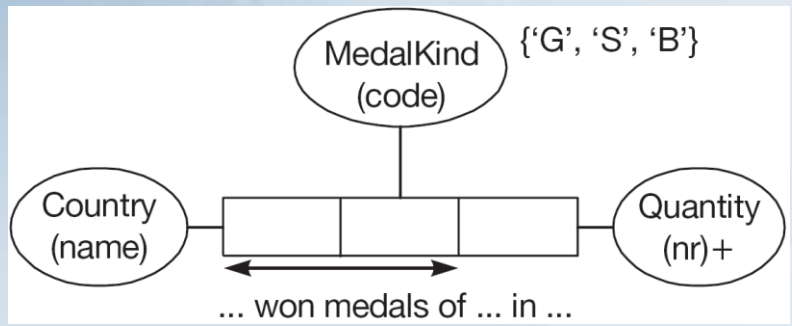


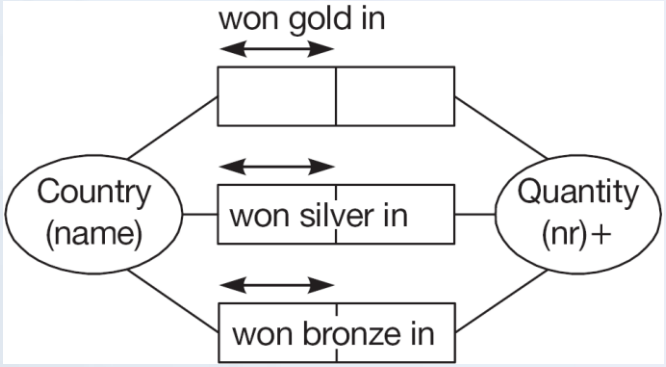
We generalize smoking and drinking into indulging in a vice, where vice has two specific cases. If we transform in the opposite direction, we specialize indulging in a vice into two predicates, one for each case.

Predicate Specialization and Generalization

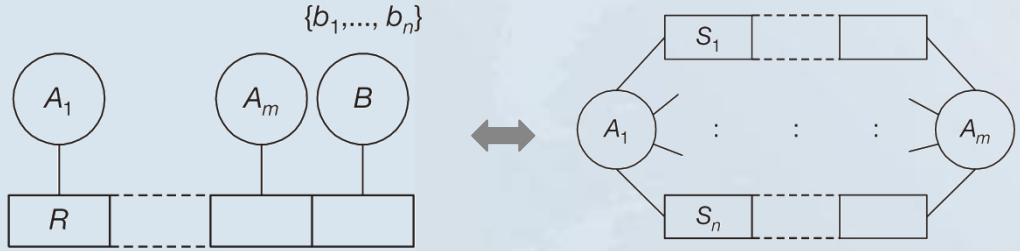
Based on [2]

If two or more predicates may be thought of as special cases of a more general predicate, then we may replace them by the more general predicate, so long as the original distinction can be preserved in some way.





Because there are exactly three kinds of medals, the ternary may be specialized into three binaries, one for each medal kind,

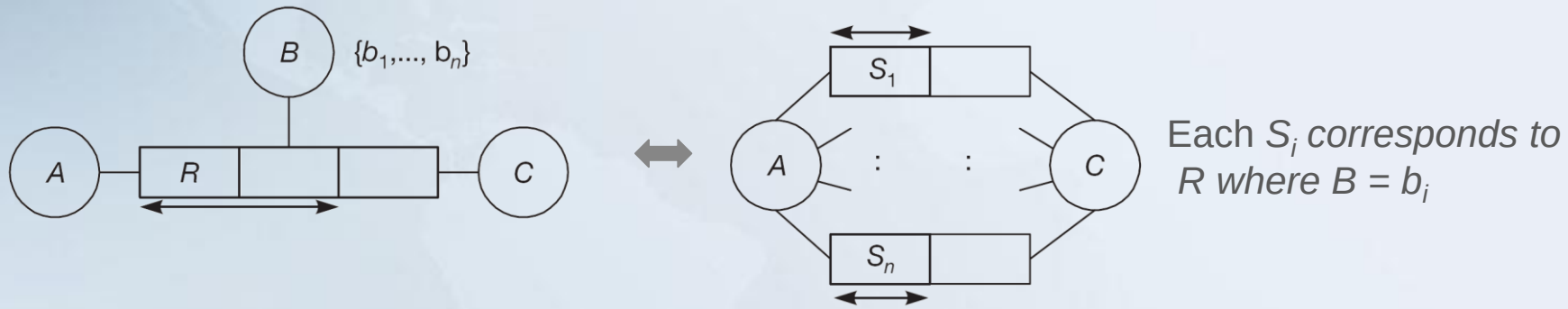


Where $m \geq 1$, and each S_i corresponds to R where $B = b_i$

Theory: *R may be specialized into $S_1..S_n$ by absorbing B.*

Predicate Specialization and Generalization

The previous theorem always holds, but any constraint added to one of the schemas must be translated into an equivalent, additional constraint on the other schema.

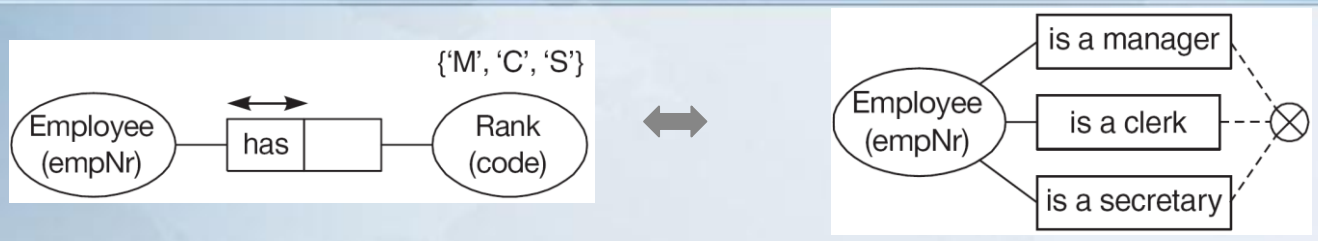


The UC on the left is equivalent to the UCs on the right.

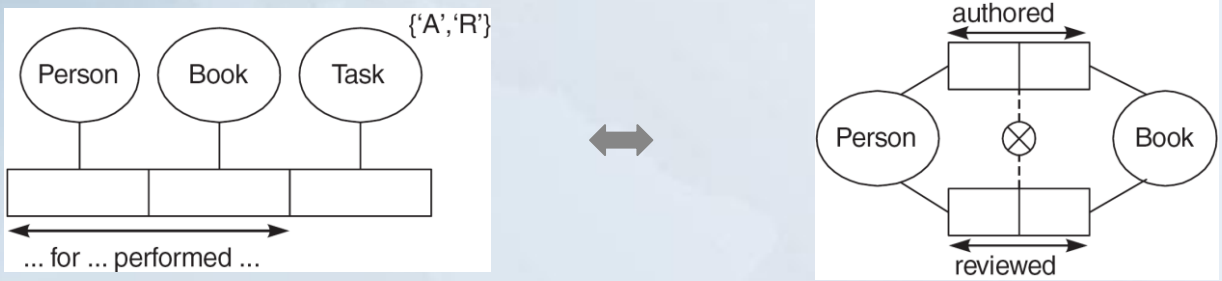
- **If a UC in R spans a combination of B's role and other roles, a UC spans the specialization of these other roles in S_1,..,S_n, and conversely.**

Predicate Specialization and Generalization

Based on [2]



The UC on the left is equivalent to the exclusion constraint on the right.



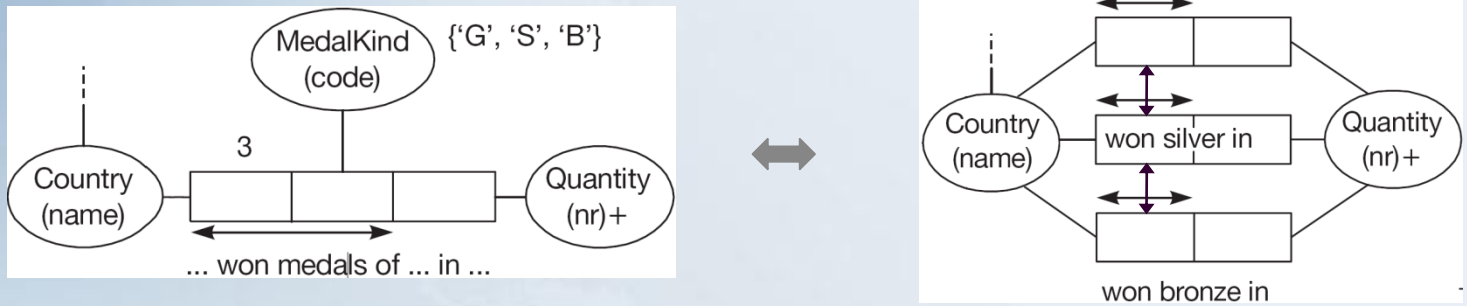
The UC on the left is equivalent to the exclusion constraint on the right.

Where $m \geq 1$, and each S_i corresponds to R where $B = b_i$

The UC on the left is equivalent to the exclusion constraint on the right.

➢**If a UC spans all roles of *R except for B's role, then* S_1 .. S_n *are mutually exclusive, and* conversely.**

Predicate Specialization and Generalization



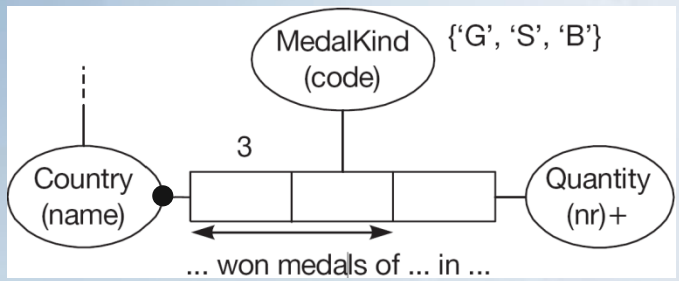
if any medal results are recorded for a country, all three medal results (gold, silver, and bronze) are required. To express, we add an *equality constraint between the medal winning roles played by Country.*

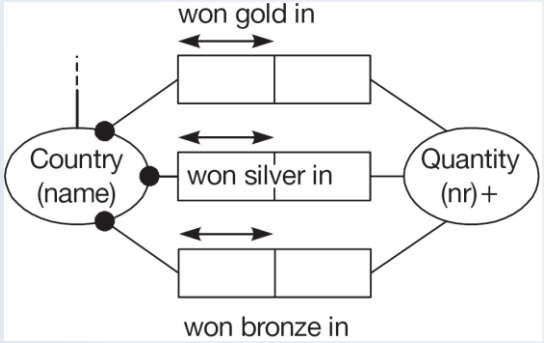
- ***If R is a ternary with a UC spanning just B's role and one other role, then adding a frequency constraint of n to this other role is equivalent to adding an equality constraint over the specialized versions of that role.***

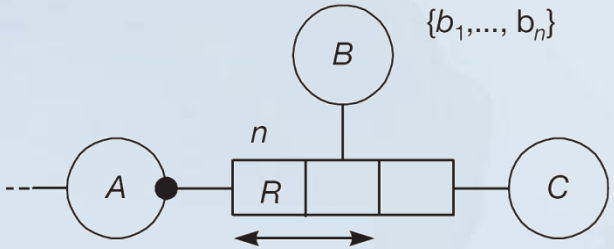
Predicate Specialization and Generalization

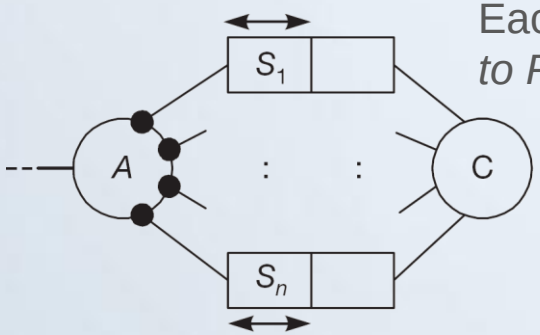
Based on [2]

The impact of adding mandatory role and frequency constraints.





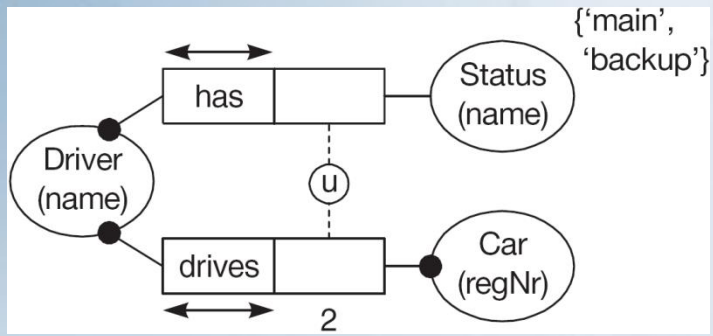


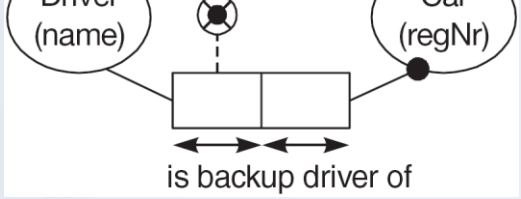


Each *S corresponds to R where B = b_i*

- ***If A's role (or role disjunction) in R is mandatory, then the disjunction of its specialized roles is mandatory, and conversely ($1 \leq i \leq m$).***
- ***If R is a ternary with a UC spanning just B's role and one other role, then adding a mandatory role constraint and frequency constraint of n (the number of possible values for B) to this other role is equivalent to making each specialized version of that role mandatory.***

Other Cases and Examples



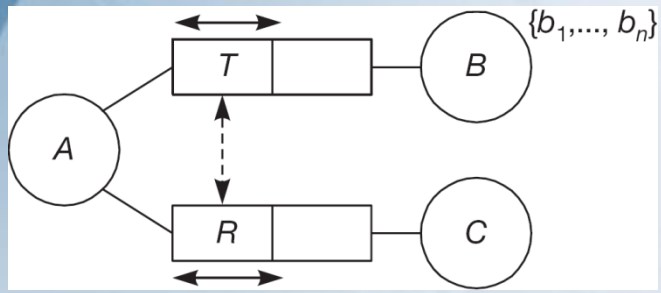


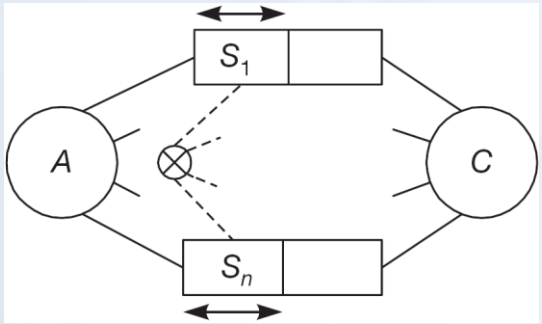
Each car in the rally has two drivers (a main driver and a backup driver), and each person drives exactly one car.

The drives predicate is specialized by absorbing Status.

Other Cases and Examples

Based on [2]





Each S_i *corresponds to R where T is restricted to* $B = b_i$

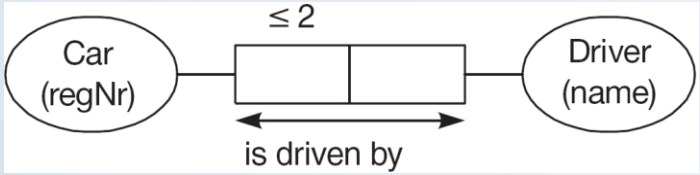
Theory: R may be specialized into S_1..S_n by absorbing B.

- **Corollary 1: If s roles are mandatory in the left-hand schema, the disjunction of s roles in the right-hand schema is mandatory, and conversely.**
- **Corollary 2: If an external UC spans the roles of and in the left-hand schema, then a UC applies to each of s roles in the right-hand schema, and conversely.**
- **Corollary 3: If s role in the left-hand schema is mandatory, then each of s roles in the right-hand schema is mandatory, and conversely.**
- **Corollary 4: An equality constraint over s roles in the RHS is equivalent to a frequency constraint of on s role in the left-hand schema; this constraint is strengthened to if a UC exists on each of s roles in the right-hand schema.**

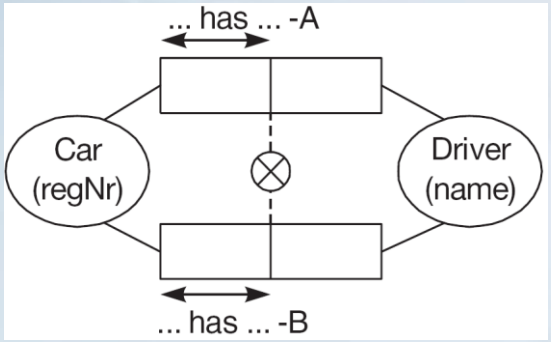
Other Cases and Examples

Based on [2]

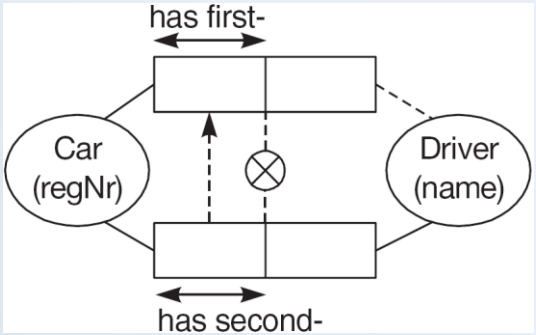
Can the predicate be specialized?



?



?

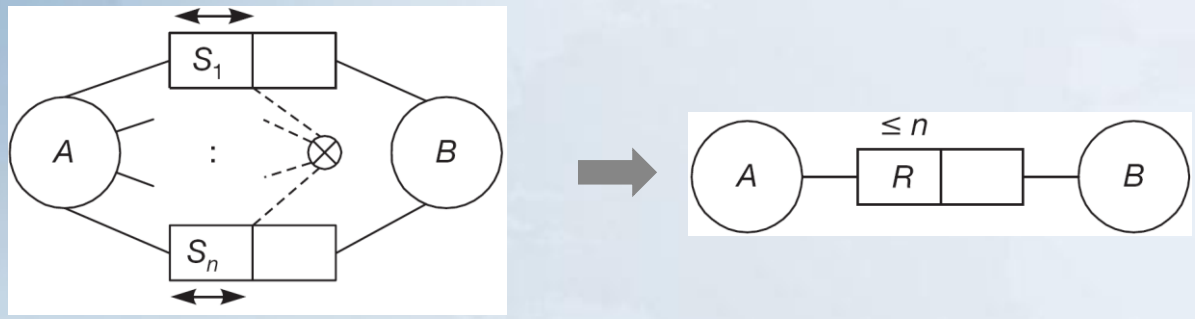


- Transforming from the original schema to one of those strengthens the schema by adding information.
- Transforming in the opposite direction weakens the schema by losing information.

➢ Any such transformations that add or lose information should be the result of conscious decisions that are acceptable to the client (for which the business domain is being modeled).

Other Cases and Examples

Based on [2]



Each S_i *corresponds to one instance of R*

Theory: The left-hand schema implies the right-hand schema.

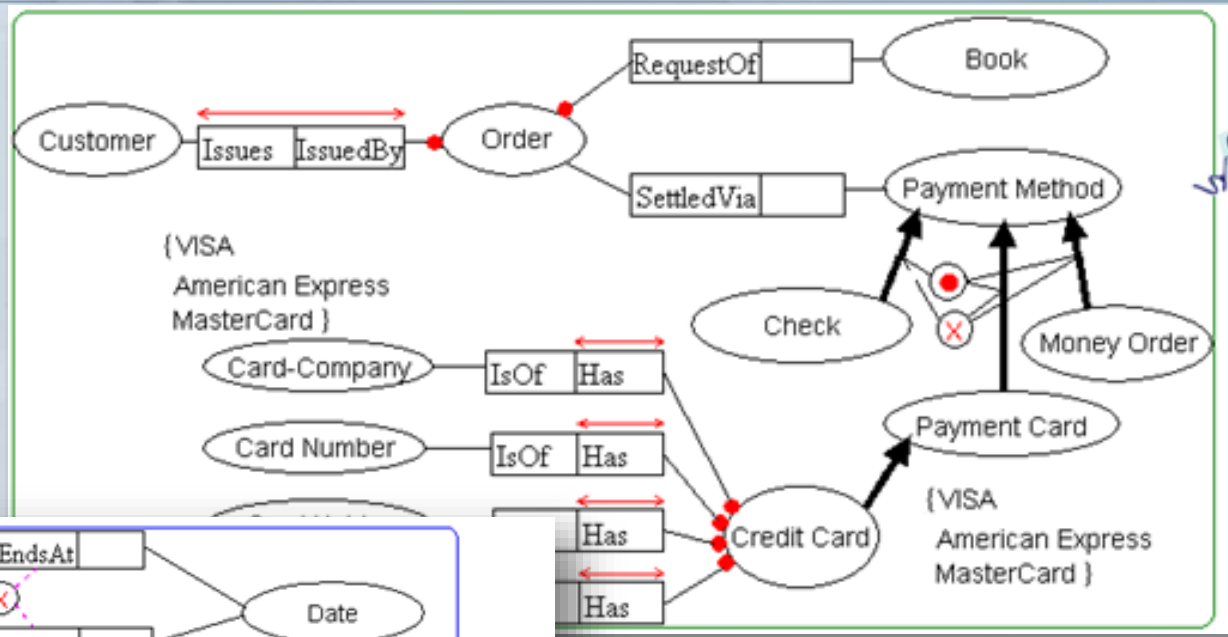
Corollary 1:If an equality constraint applies over s roles in the left-hand schema, then the frequency constraint in the right-hand schema is strengthened to , and conversely.

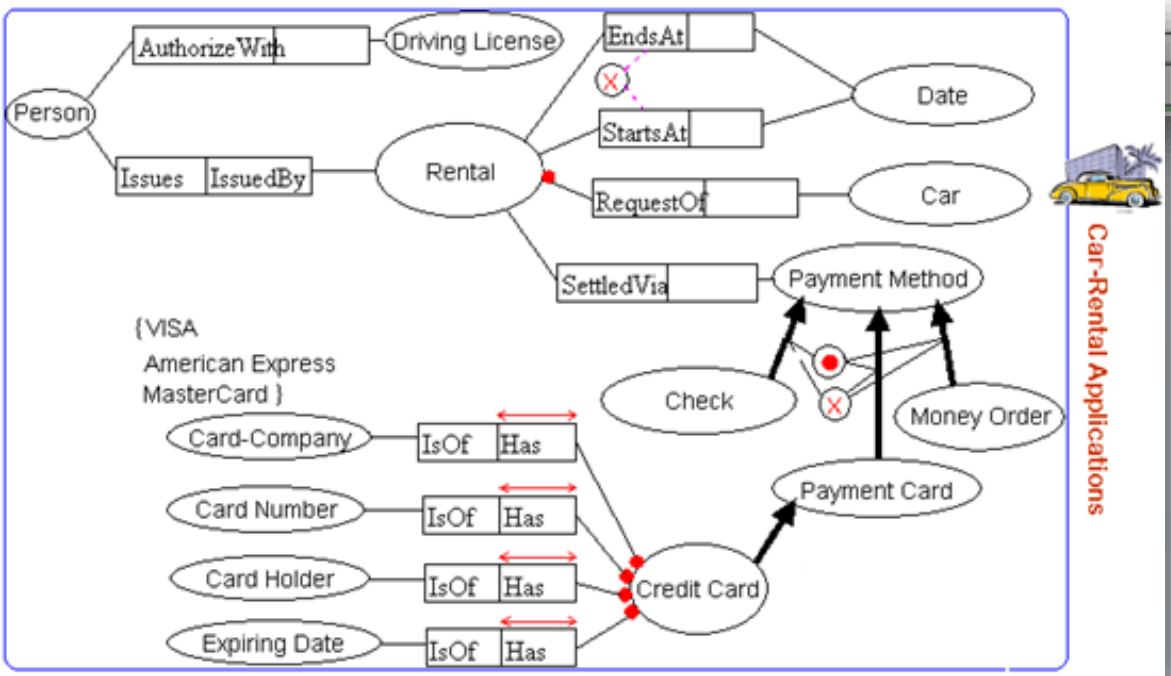
Corollary 2: Adding a UC to role in the right-hand schema is equivalent in the left-hand schema to adding UCs to s roles (making the S 1:1) and strengthening the exclusion constraint to an exclusion constraint over s roles.

Outline

- Final Check
  - Rules Implications
  - Rules Contradictions
  - Modeling Tips (Check List)
- Rules Verbalization
- Schema Equivalence and Optimization
- Schema Modularization

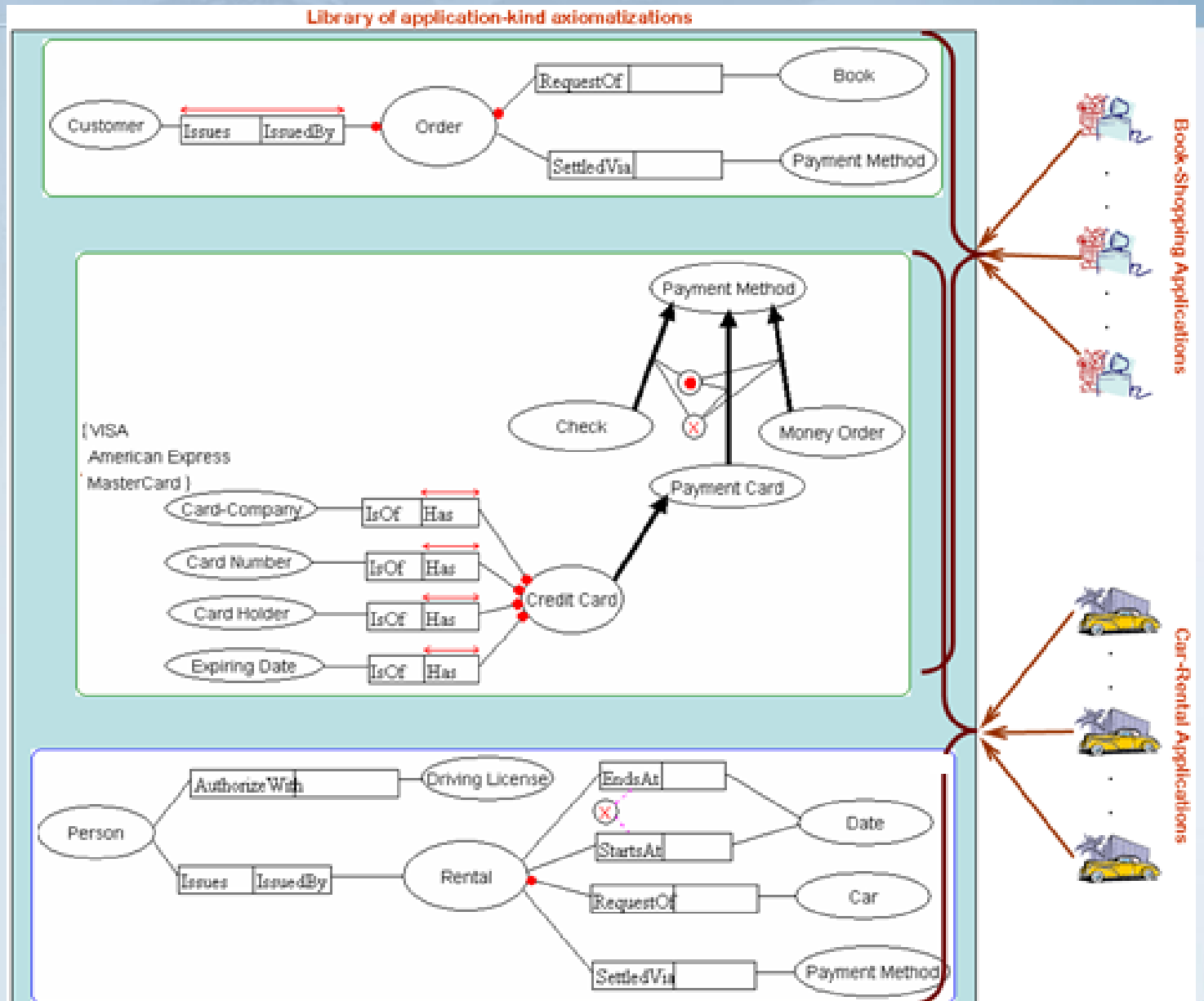
Modularization



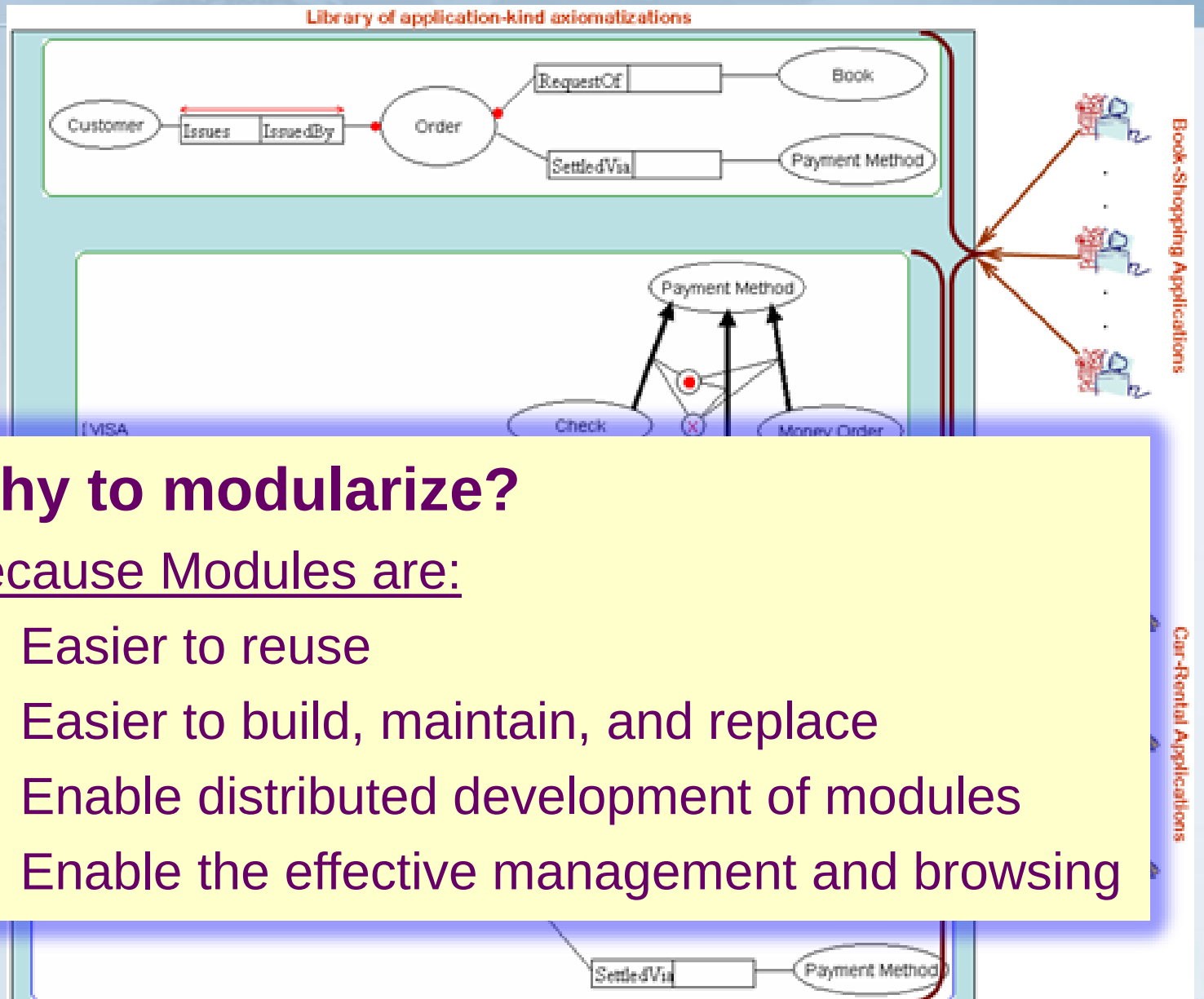


- Develop a conceptual schema as a set of modules and later compose to form one module.

Modularization



Modularization

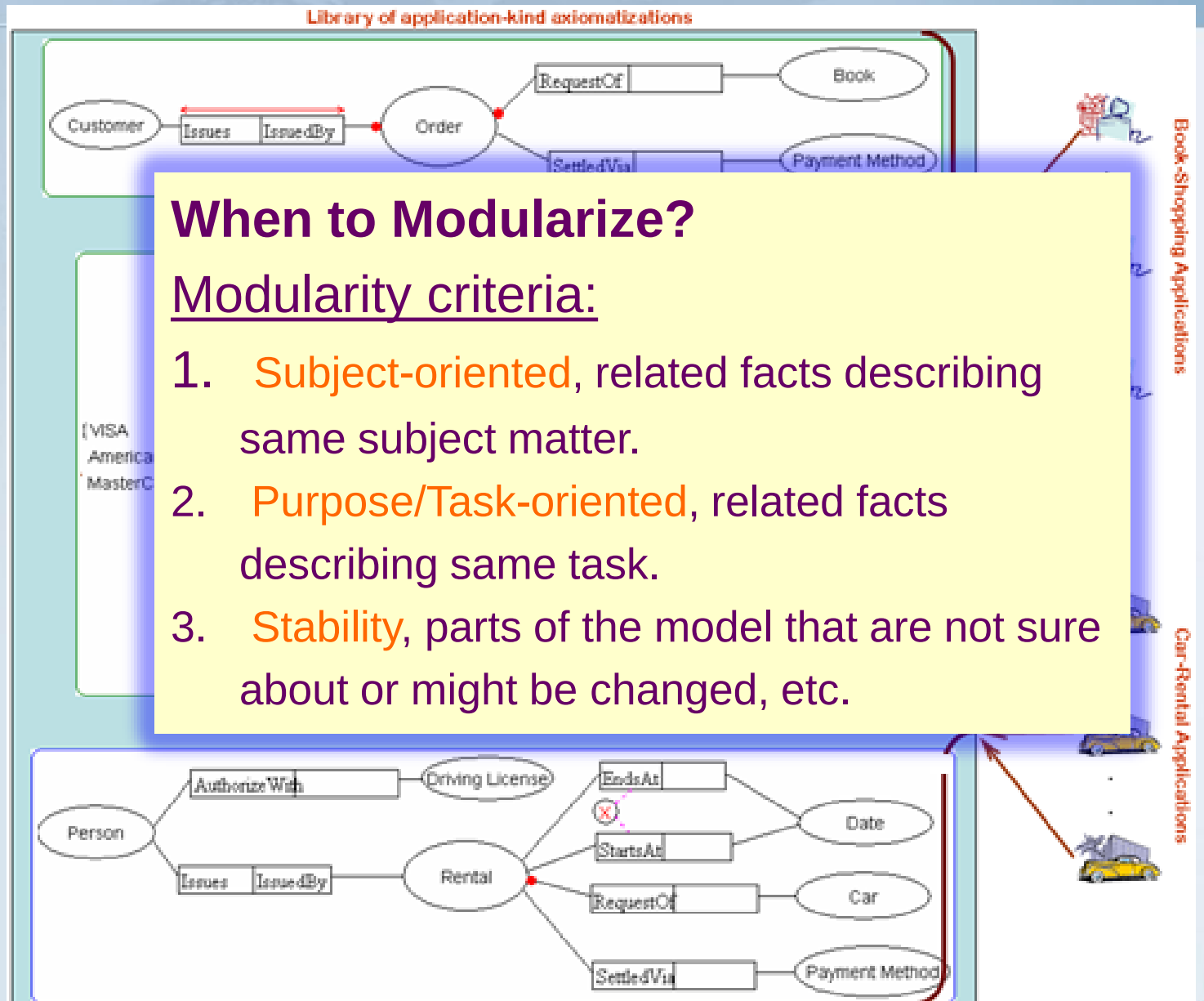


Why to modularize?

Because Modules are:

1. Easier to reuse
2. Easier to build, maintain, and replace
3. Enable distributed development of modules
4. Enable the effective management and browsing

Modularization



When to Modularize?

Modularity criteria:

1. Subject-oriented, related facts describing same subject matter.
2. Purpose/Task-oriented, related facts describing same task.
3. Stability, parts of the model that are not sure about or might be changed, etc.

References

1. Jarrar, Mustafa, and Stijn Heymans. "Towards Pattern-based Reasoning for Friendly Ontology Debugging." Print.
2. Halpin, T. A., and Tony Morgan. *Information Modeling and Relational Databases*. 2nd ed. Burlington, MA: Morgan Kaufmann, 2008. Print.
3. Jarrar, Mustafa, C. Maria Keet, and Paolo Dongilli. "Multilingual Verbalization of ORM Conceptual Models and Axiomatized Ontologies." Print.