Consistency in the Business System Model

Václav Řepa
University of Economics
Prague, Czech Republic

Consistency in the Business System Model

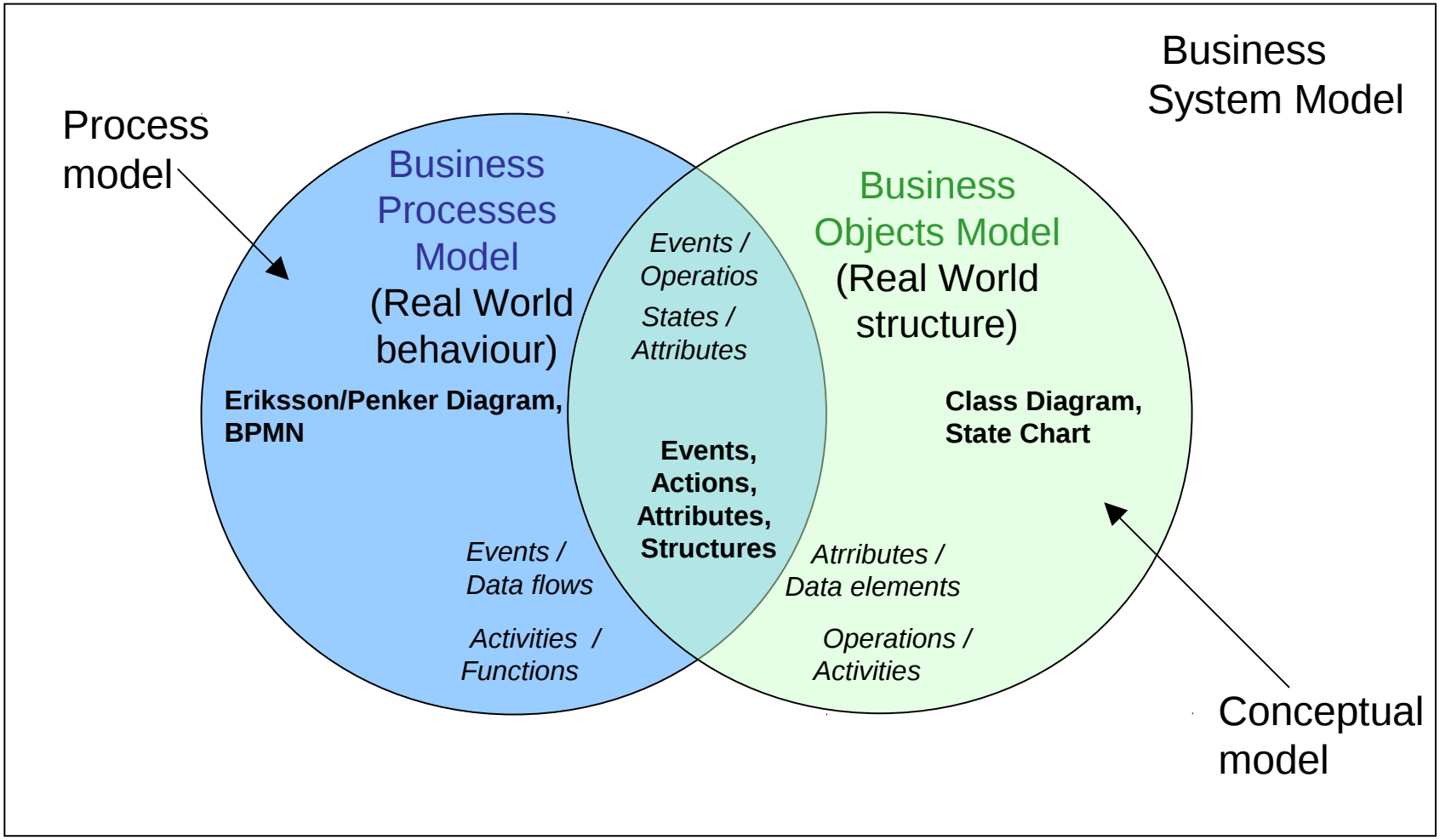
Agenda

1. Modelling the Business system
   - Two Basic Dimensions of the Real World
   - Business Processes versus Class Life Cycles
2. Consistency Rules
   - Completeness
   - Correctness
3. Structural Consistency
   - M.A.Jackson
   - Structural Consistency Rules
4. Summary

Presented work is a part of the project *OpenSoul*.
The project is aimed on the development of the Business System Modeling Methodology based on the formal meta-model.
For more information see http://opensoul.panrepa.org/.

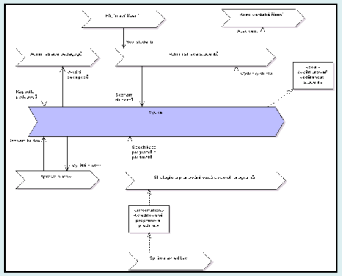
1. Modelling the Business System

Two Parts of the Real World (Business System) Model

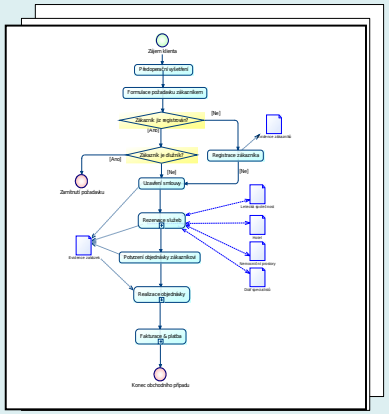


Business System Analytical Models – an overview -

Business Processes
(Global process model, Process Diagrams)



Eriksson-Penker Notation



Business Process Modeling Notation

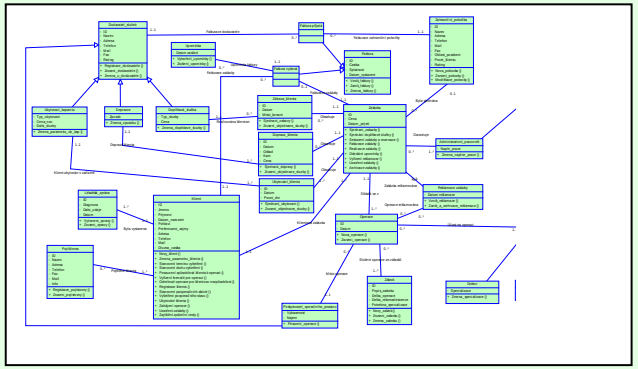
Products, inputs, outputs, actors, business restrictions (objects life cycles)

Intentional combinations of objects life cycles, context of objects lives

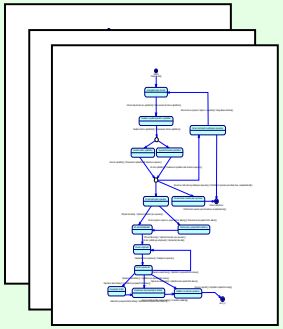
Events and their general context

Intentional combinations of events

Business Objects
(Class Diagram, State Charts)



Class Diagram



State Chart

Unified Modeling Language

Relations between the Process Model and the Object Model

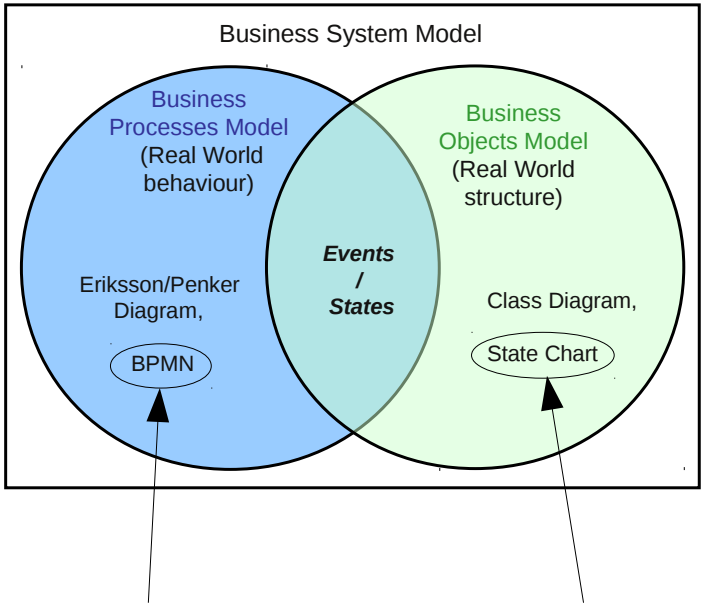
Process model explains dependencies between objects and their life cycles giving them the superior sense (intention):

- events and process states are the reasons for object actions,
- objects are playing roles of attendees or victims (subjects) of processes.

One object typically occurs in more processes as well as one process typically combines more objects
$\Rightarrow$ two orthogonal, conceptually different processes:

- process of object's life (life cycle) - combination of actions of various business processes,
- business process - combination of actions of various objects.

Dynamics of the Real World



Intentional Dynamics (**behaviour** in the Real World)

Structural Dynamics (Real World dynamics **rules**)

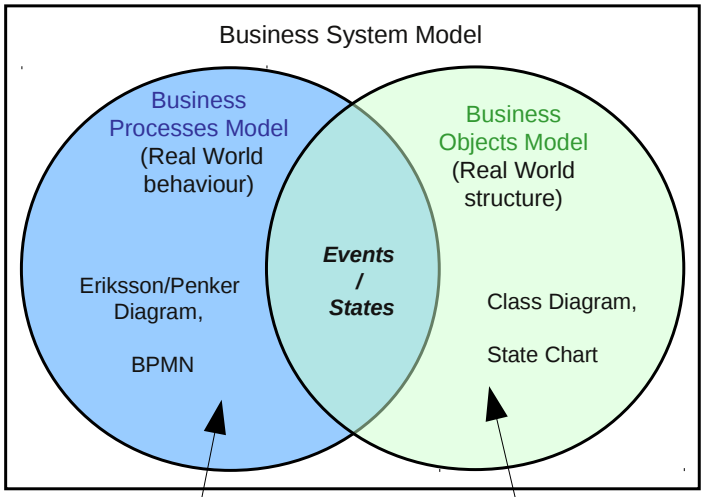
Real World dynamics belongs not to the Business processes only, but also to the Business Objects (conceptual) Model - in the form of **objects life cycles** (ordering of methods in one algorithm).

Such dynamics is seen from the point of view of objects and their relationships and **should not be regarded as behaviour**

(it is not driven by any intention, has no goal nor product).

Thus the "***behaviour of objects***" *should be regarded as a* ***structural aspect of the Real World***.

Dynamics of the Real World



Dynamic Quality (Pirsig)

Static Quality (Pirsig)

Real World dynamics belongs not to the Business processes only, but also to the Business Objects (conceptual) Model - in the form of **objects life cycles** (ordering of methods in one algorithm).

Such dynamics is seen from the point of view of objects and their relationships and **should not be regarded as behaviour**

(it is not driven by any intention, has no goal nor product).

Robert Pirsig: Metaphysics of Quality

R.M.Pirsig: Zen and the Art of Motorcycle Maintenance: an Inquiry into Values, William Morrow, 1974

R.M.Pirsig: Lila: an Inquiry into Morals, Bantam Press, 1991

2. Consistency Rules

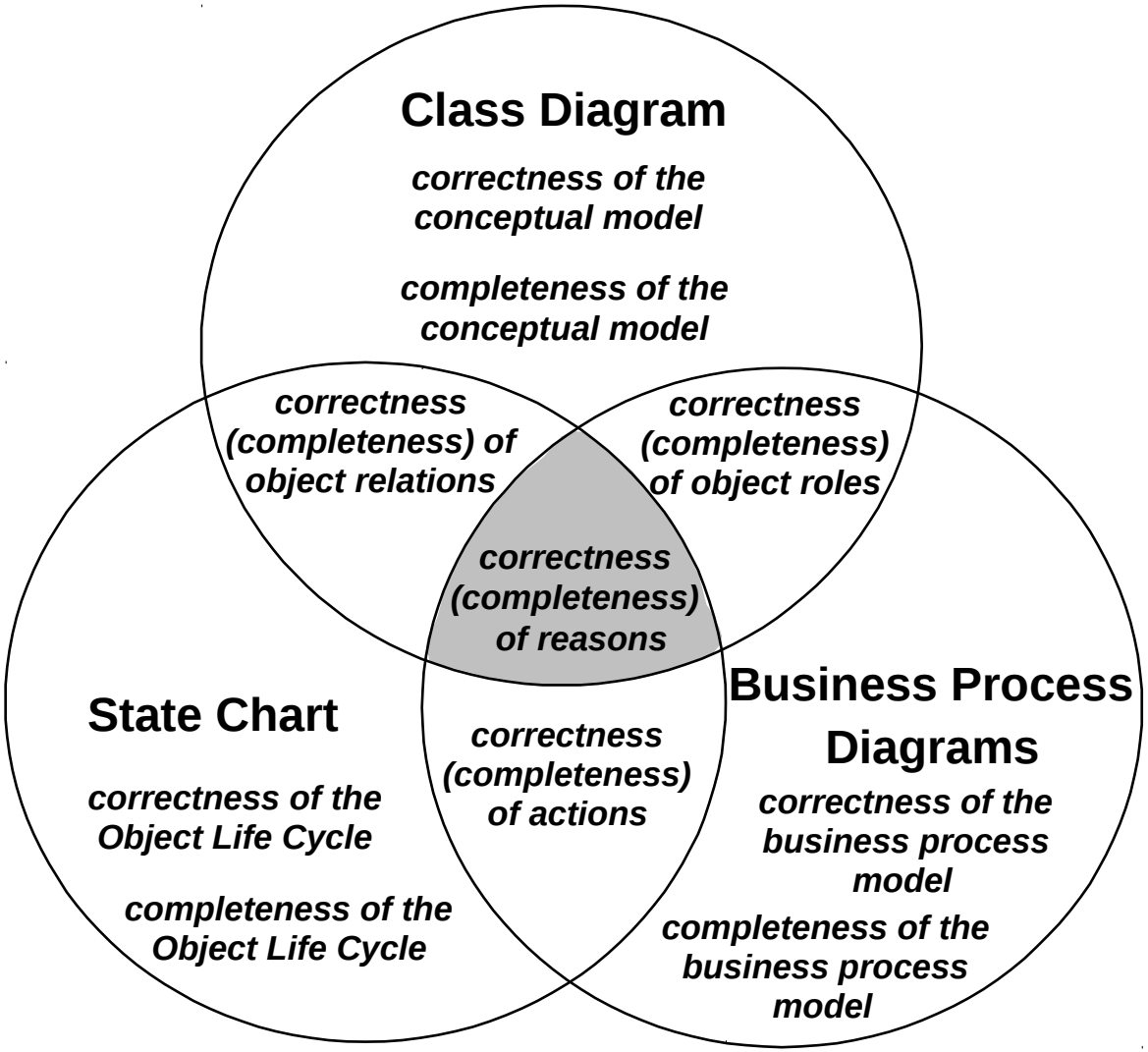
Consistency

The particular set of models is regarded as consistent if there is **no contradiction among different kinds of the same information** expressed in the models.

- Contradiction can occur if there are different ways of expressing the same information in different models.
- Different views on the same fact is a *basic working tool* of the methodology.
  - it stimulates the creativity of analyser,
  - it allows uncovering the facts which are hidden in other facts from one point of view but visible from the other point of view.

The concept of Consistency represents not only a ***danger*** (of inconsistency) but an ***opportunity*** (to improve the quality of models) as well.

Coherency of models - overview



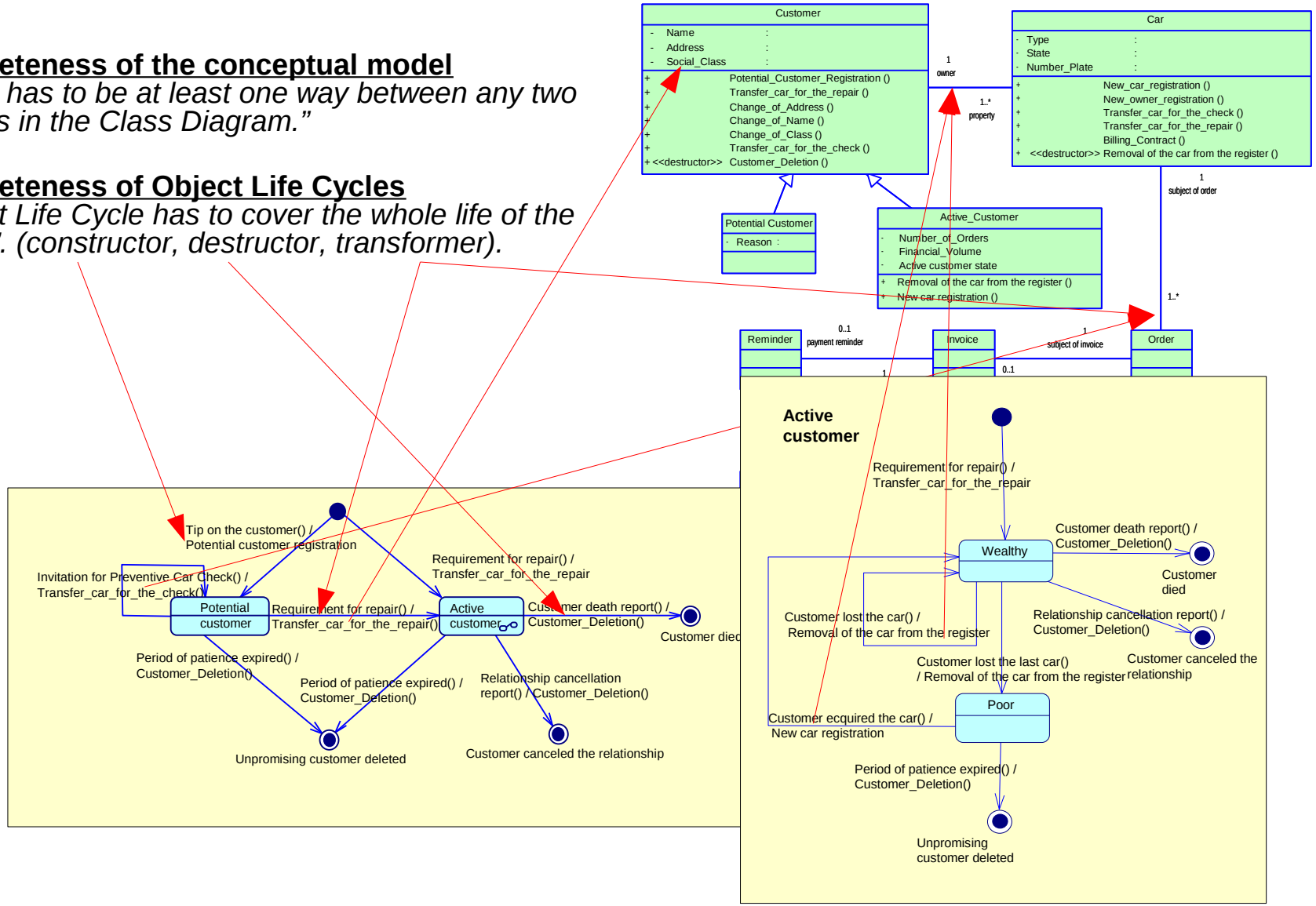
Criteria of completeness

- **completeness of the conceptual model** generally follows from the theory of conceptual modeling, where the basic rules for this criterion are defined. For instance:
  - "There has to be at least one way between any two classes in the Class Diagram.",
- **completeness of the business process model** generally follows from the theory of business processes reengineering and modeling, where the content of this concept in the field of business processes is defined. For instance:
  - "There has to be business process model described for each specified product."
  - "Each recognized event has to be used in at least one business process model as a reason for some action.
    (by the way: this rule defines the objective need for decomposition of processes – we need to decompose processes until we place all events)".), etc.
- **completeness of Object Life Cycles** is expressed by the simple rule:
  - "Object Life Cycle has to cover the whole life of the object".
    As a realization of this rule the methodology defines three mandatory types of object methods (stereotypes): constructor, destructor, and transformer. The purpose is to ensure the completeness of the whole object life in the description.

Example of the completeness of models

completeness of the conceptual model
"There has to be at least one way between any two classes in the Class Diagram."

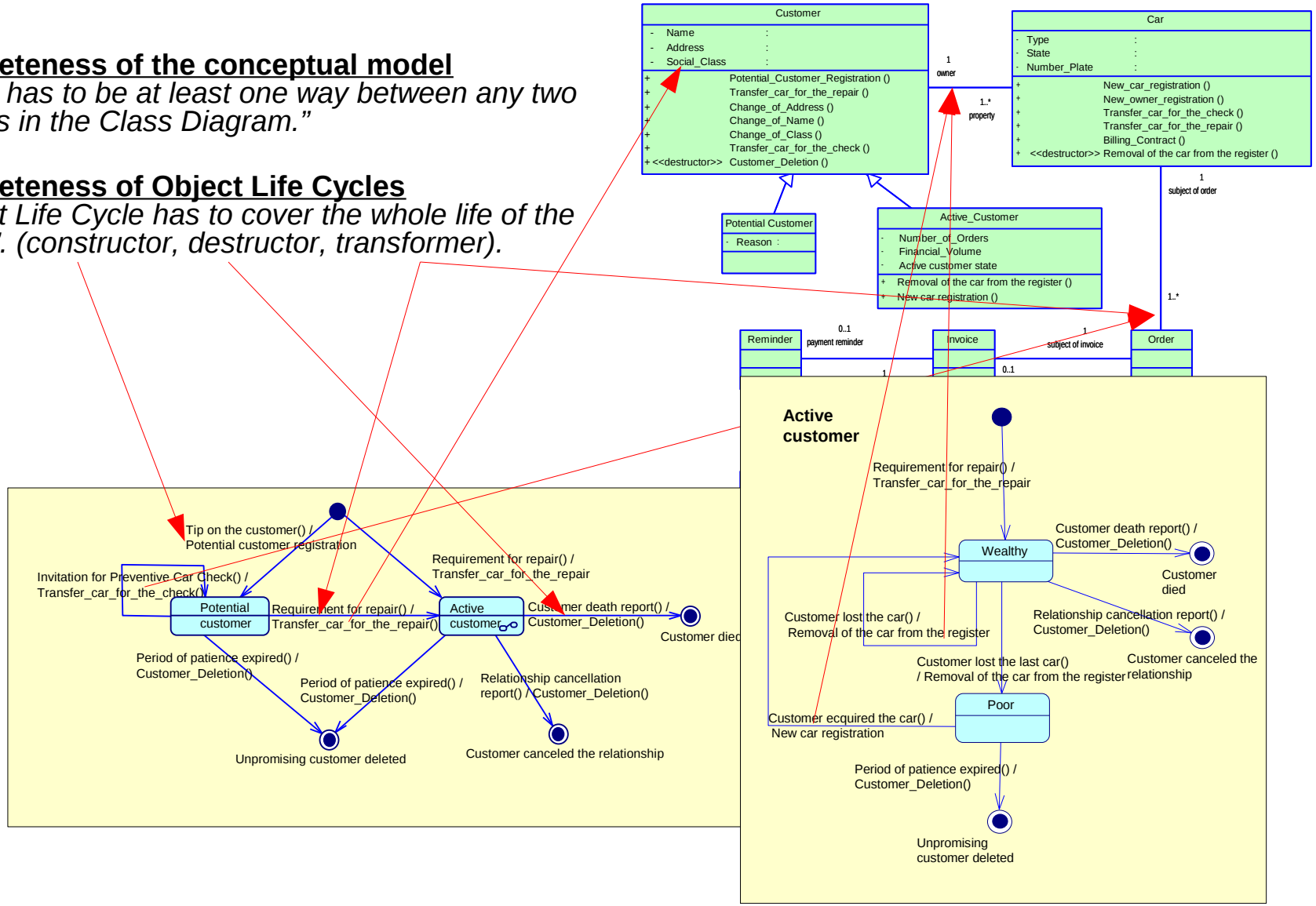
completeness of Object Life Cycles
"Object Life Cycle has to cover the whole life of the object". (constructor, destructor, transformer).



Example of the completeness of models

completeness of the conceptual model
"There has to be at least one way between any two classes in the Class Diagram."

completeness of Object Life Cycles
"Object Life Cycle has to cover the whole life of the object". (constructor, destructor, transformer).



Criteria of correctness

- **correctness of the conceptual model** is defined as follows:
  - “Each object class has to correspond to the real and existing objects. Any relationship to other object class(es) has to model the existing possible relationship. Described object classes and their relationships have to be valid for all possible instances of each object class.”,
- **correctness of the business process model** is defined as follows:
  - “Business process has to fulfill the main process goal. Described process actions, their succession, inputs, outputs and other attributes have to be valid for all possible instances of the process.”,
- **correctness of Object Life Cycles** is defined as follows:
  - “Object Life Cycle has to correspond to the real and objective actions and their successions in the life of the object. Object Life Cycle has to be valid for all possible instances of the object class.”.

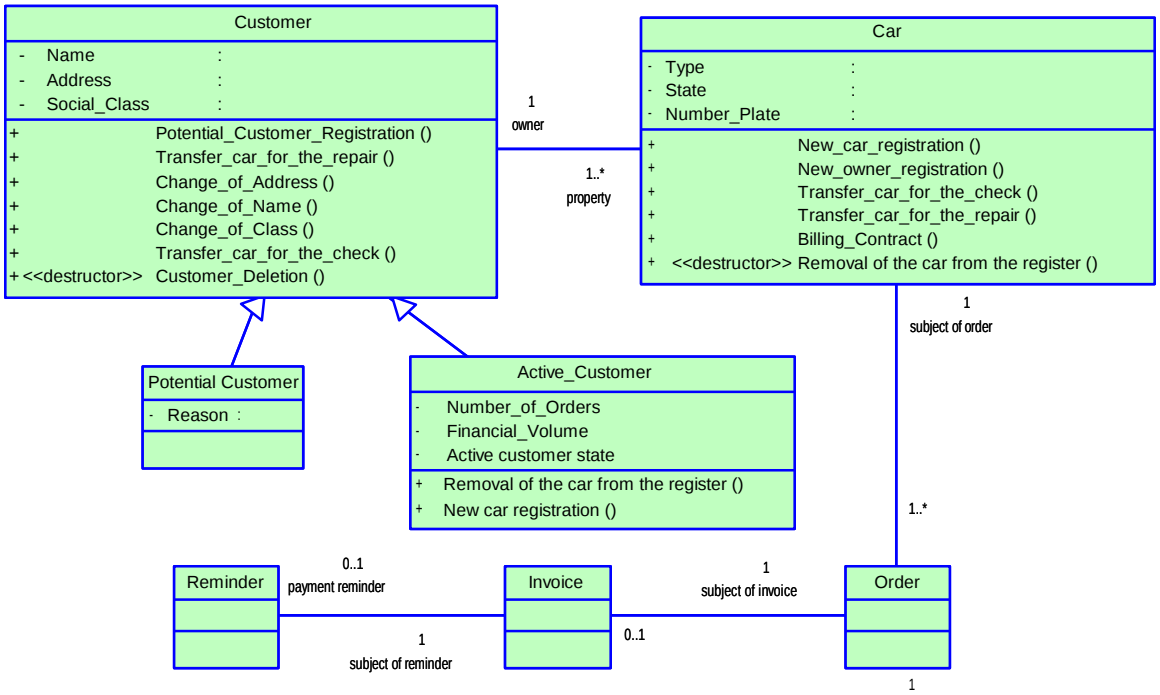
Criteria of correctness&completeness

- **correctness (completeness) of object relations**
  relationships between State Chart and Class Diagram:
  - "Each association belonging to the class in the Class Diagram has to correspond to some method specified in the object life cycle (State Chart) of this class as an attribute of the state transition, and vice versa",
- **correctness (completeness) of object roles**
  relationships between Class Diagram and Business Process Diagram:
  - "Each object class has to be present in some Business Process as an Input, or Output Set, Actor or any other external factor, and vice versa.",
- **correctness (completeness) of actions**
  relationships between Business Process Diagram and State Chart :
  - "Each action in each business process has to correspond to at least one transition between states in at least one object life cycle, and vice versa."
- **correctness (completeness) of reasons**
  relationships among all three diagrams:
  - "Each event used in each Object Life Cycle as a reason for the state transition should correspond to the same event used in at least one Business Processes as a reason for the process activity, and vice versa.".

Example of the correctness (completeness)

Correctness (completeness) of object relations
relationships between State Chart and Class Diagram:

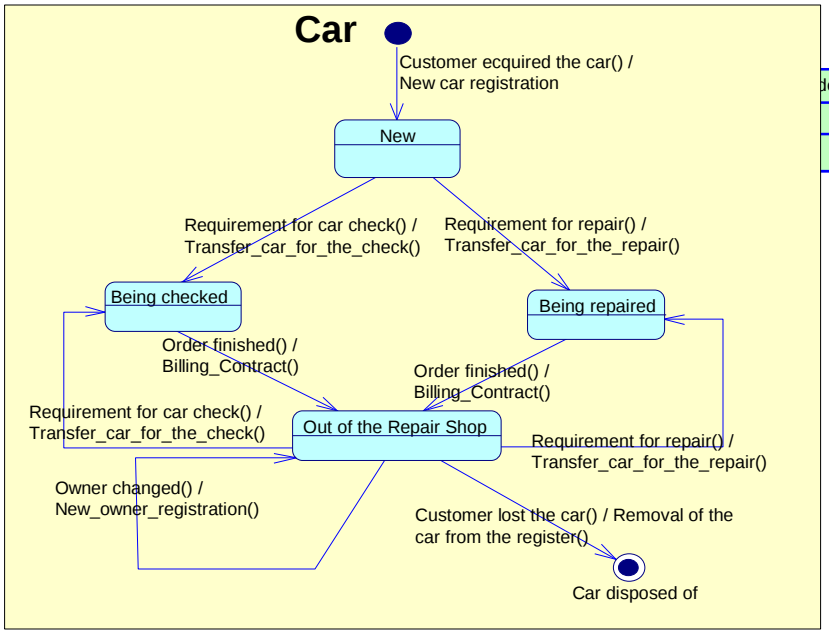
"Each association belonging to the class in the Class Diagram has to correspond to some method specified in the object life cycle (State Chart) of this class as an attribute of the state transition, and vice versa"

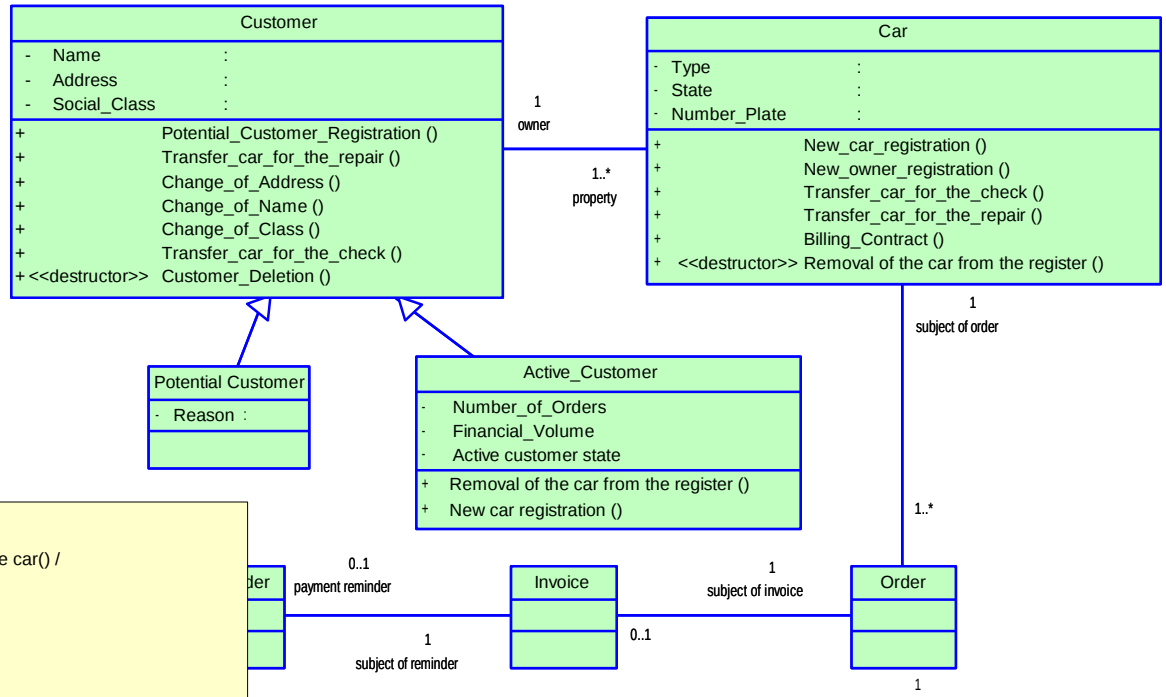


Example of the correctness (completeness)

Correctness (completeness) of object relations relationships between State Chart and Class Diagram:

"Each association belonging to the class in the Class Diagram has to correspond to some method specified in the object life cycle (State Chart) of this class as an attribute of the state transition, and vice versa"

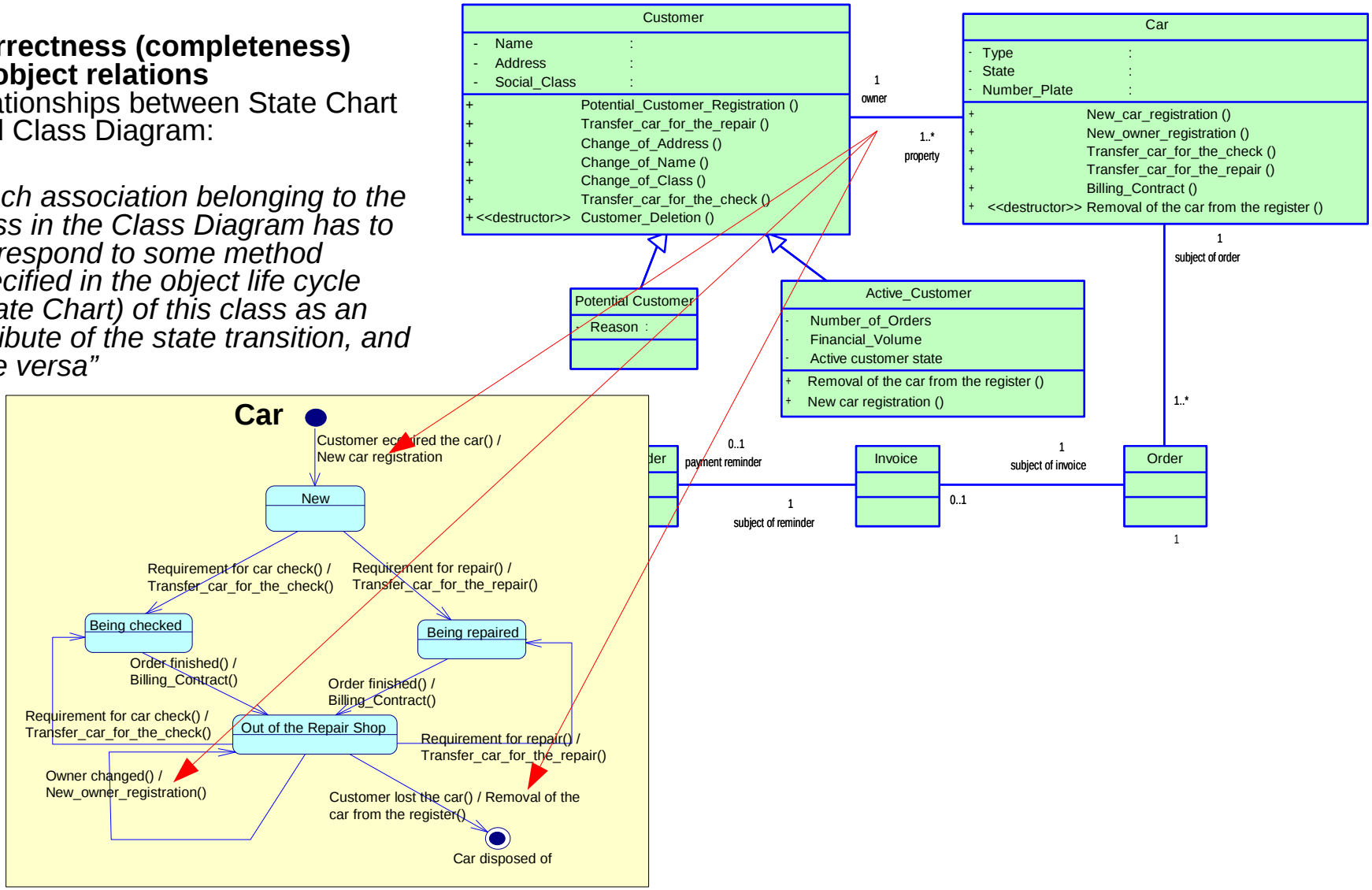




Example of the correctness (completeness)

Correctness (completeness) of object relations
relationships between State Chart and Class Diagram:

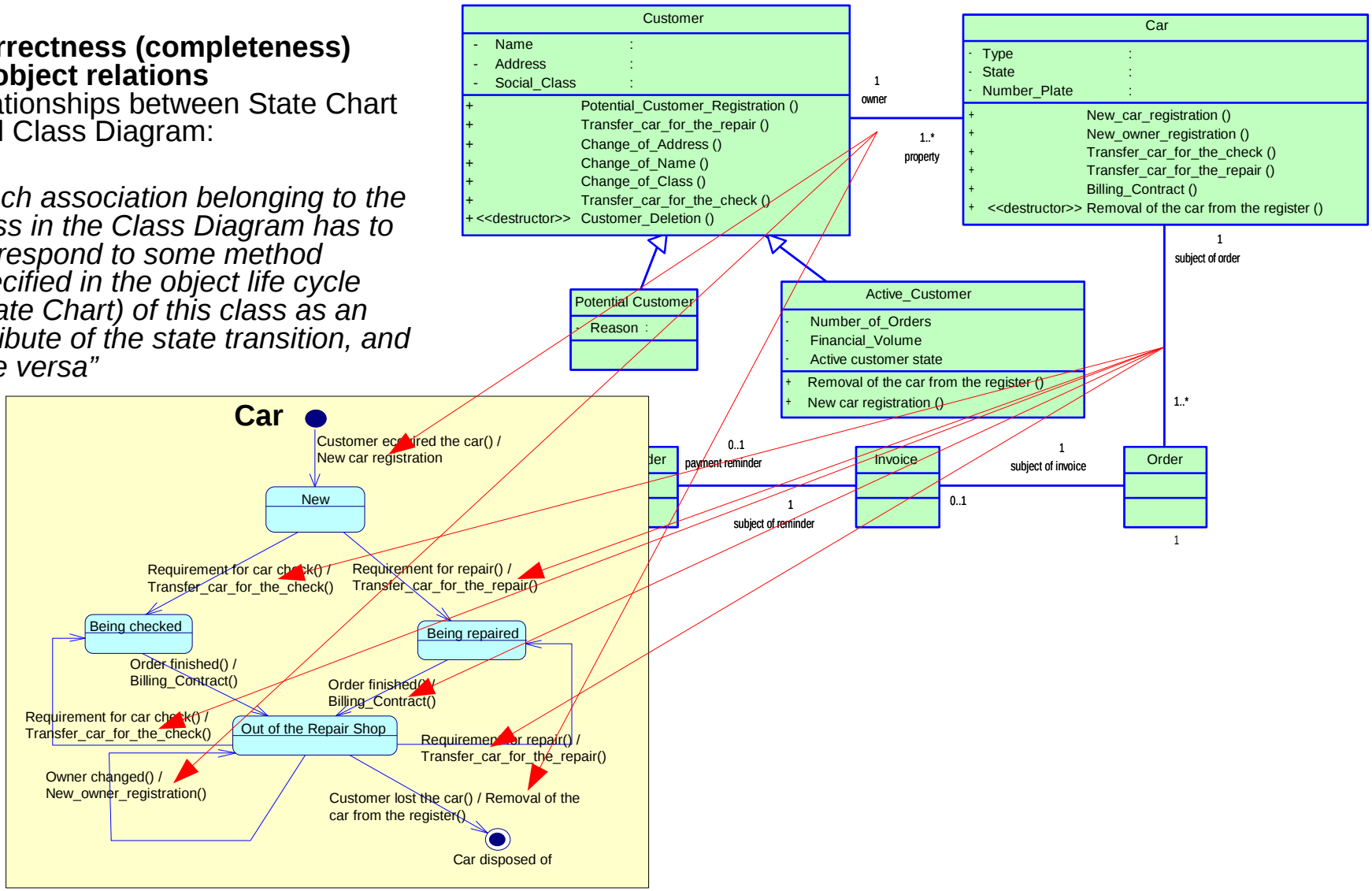
"Each association belonging to the class in the Class Diagram has to correspond to some method specified in the object life cycle (State Chart) of this class as an attribute of the state transition, and vice versa"



Example of the correctness (completeness)

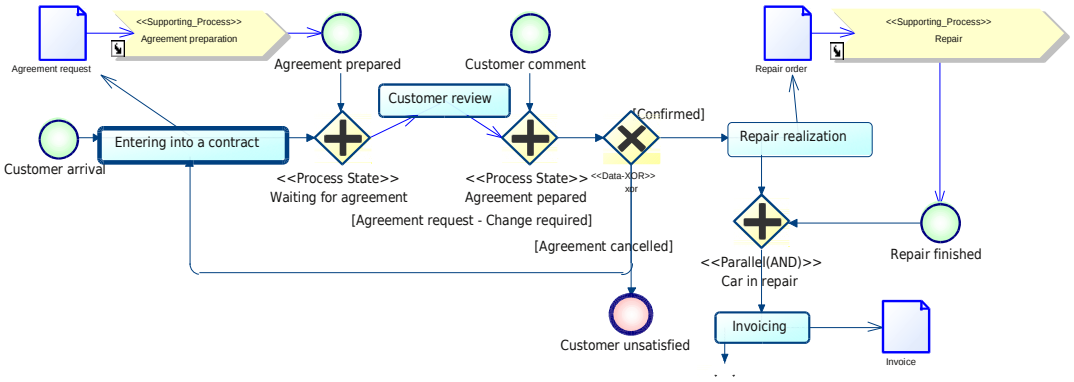
Correctness (completeness) of object relations relationships between State Chart and Class Diagram:

"Each association belonging to the class in the Class Diagram has to correspond to some method specified in the object life cycle (State Chart) of this class as an attribute of the state transition, and vice versa"



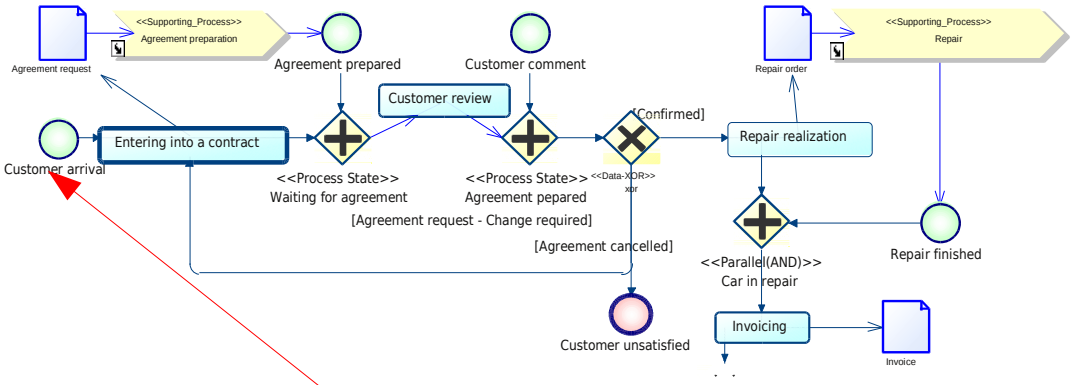
Example of the correctness (completeness)

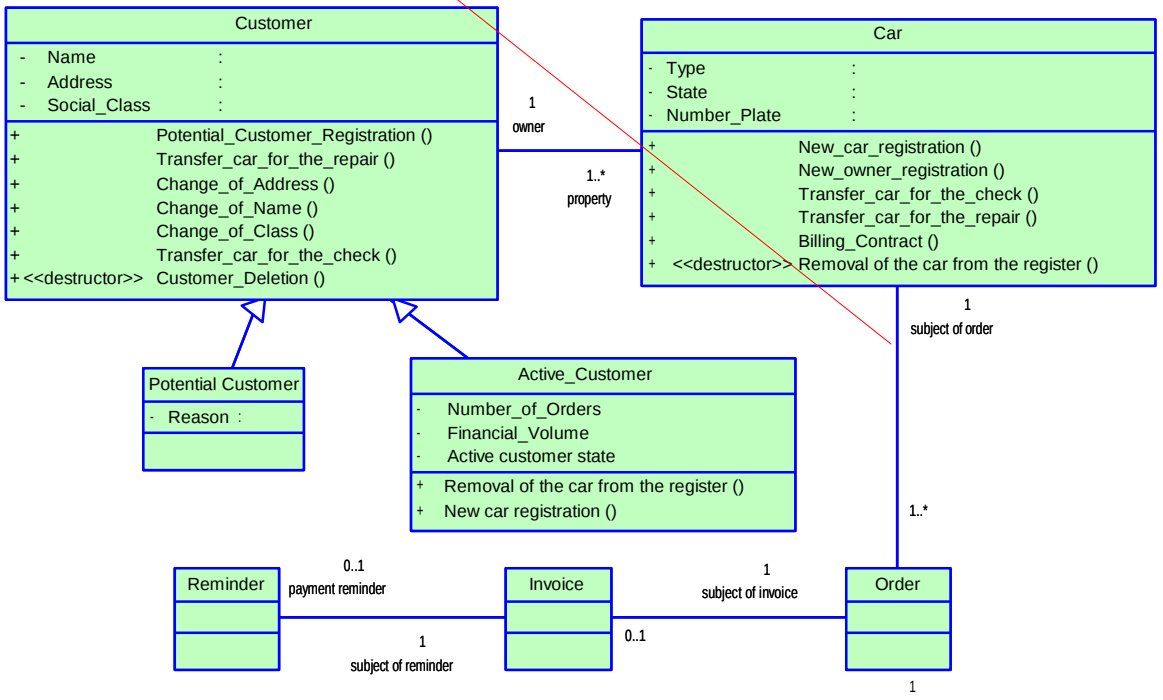
correctness (completeness) of reasons
“Each event used in each Object Life Cycle as a reason for the state transition should correspond to the same event used in at least one Business Processes as a reason for the process activity, and vice versa.”



Example of the correctness (completeness)

correctness (completeness) of reasons
"Each event used in each Object Life Cycle as a reason for the state transition should correspond to the same event used in at least one Business Processes as a reason for the process activity, and vice versa."

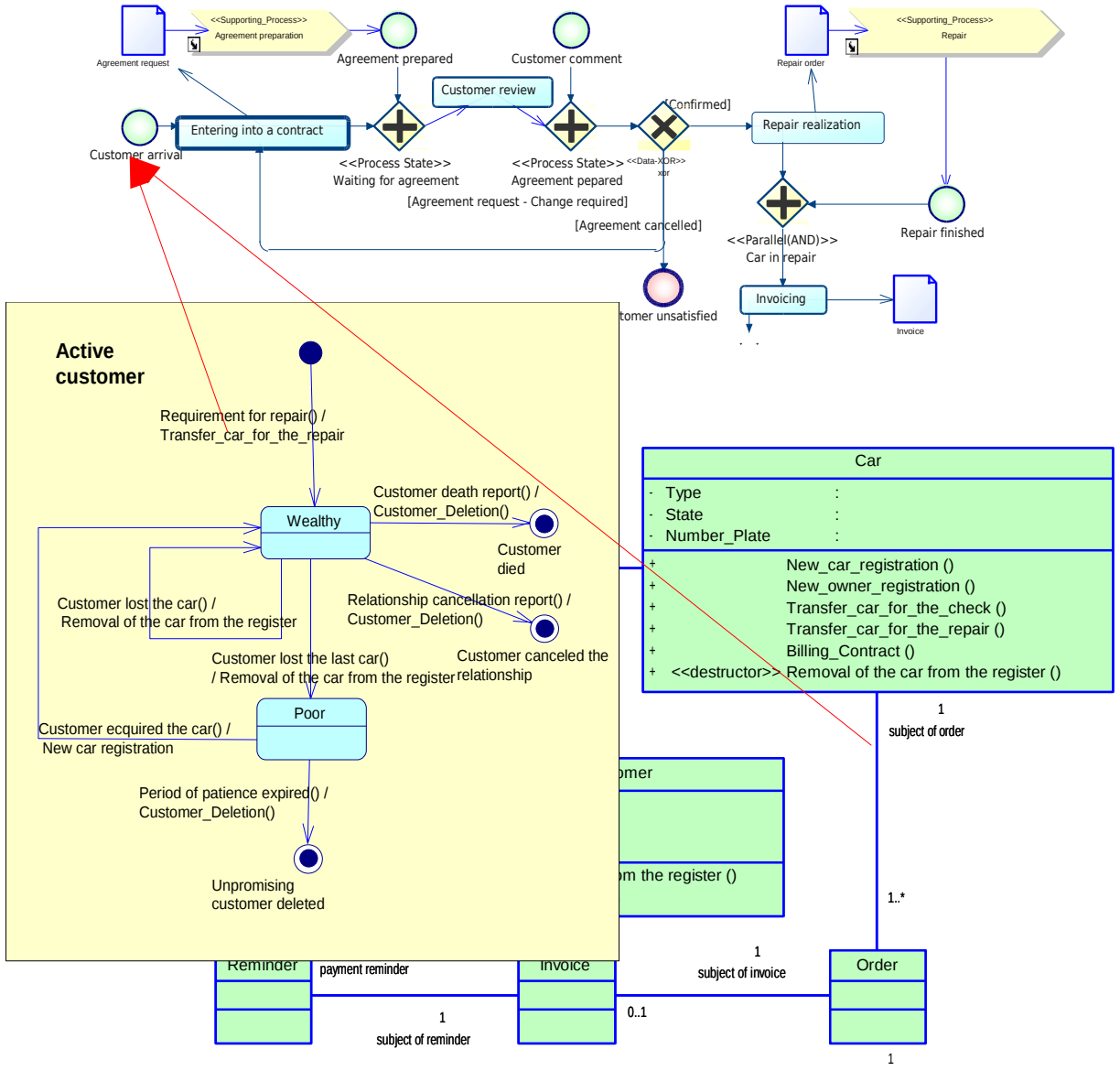




Example of the correctness (completeness)

correctness (completeness) of reasons

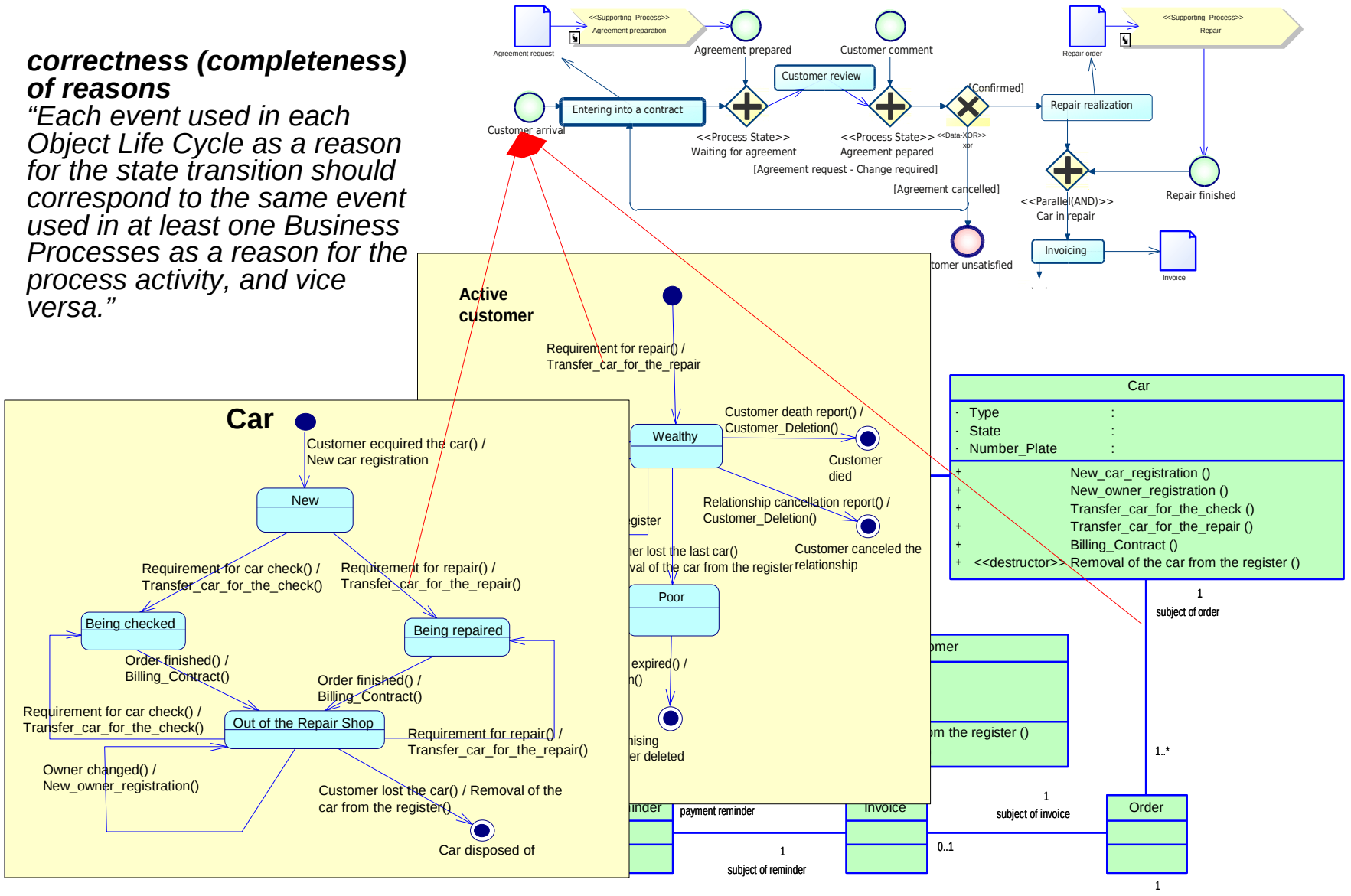
"Each event used in each Object Life Cycle as a reason for the state transition should correspond to the same event used in at least one Business Processes as a reason for the process activity, and vice versa."



Example of the correctness (completeness)

correctness (completeness) of reasons

"Each event used in each Object Life Cycle as a reason for the state transition should correspond to the same event used in at least one Business Processes as a reason for the process activity, and vice versa."



3.
Structural Consistency

Structural consistency

Structure
is a type of ordering of the group of elements.

Structural consistency
is a correspondence of structures in different models which represent different views on the same content.

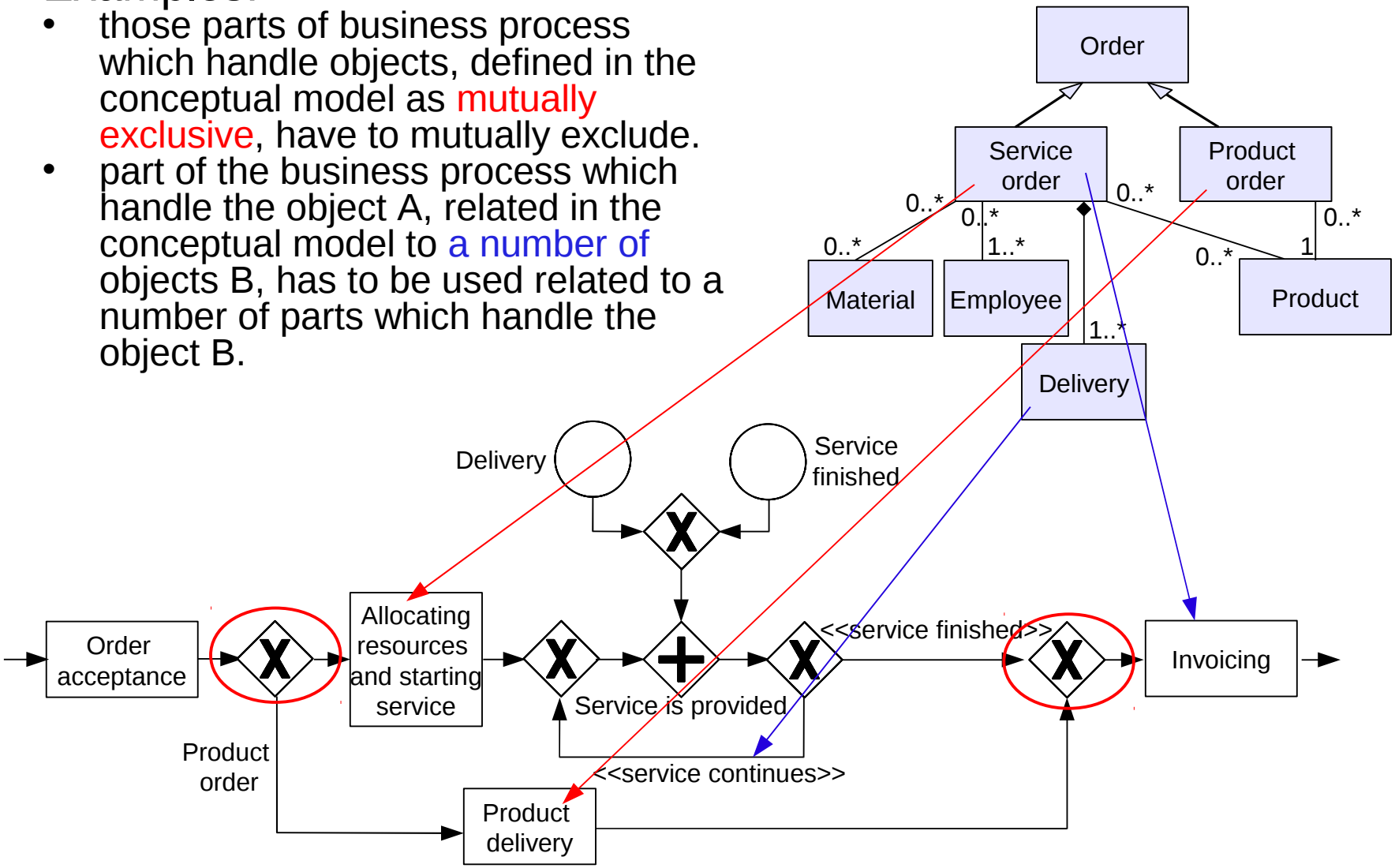
- *group of elements from one model which corresponds to the group of elements from the other model **by content** has to correspond to it **by the type of structure** as well.*

Examples:
- those parts of business process which handle objects, defined in the conceptual model as mutually exclusive, have to mutually exclude.
- part of the business process which handle the object A, related in the conceptual model to a number of objects B, has to be used related to a number of parts which handle the object B.
- ...

Structural consistency

Examples:

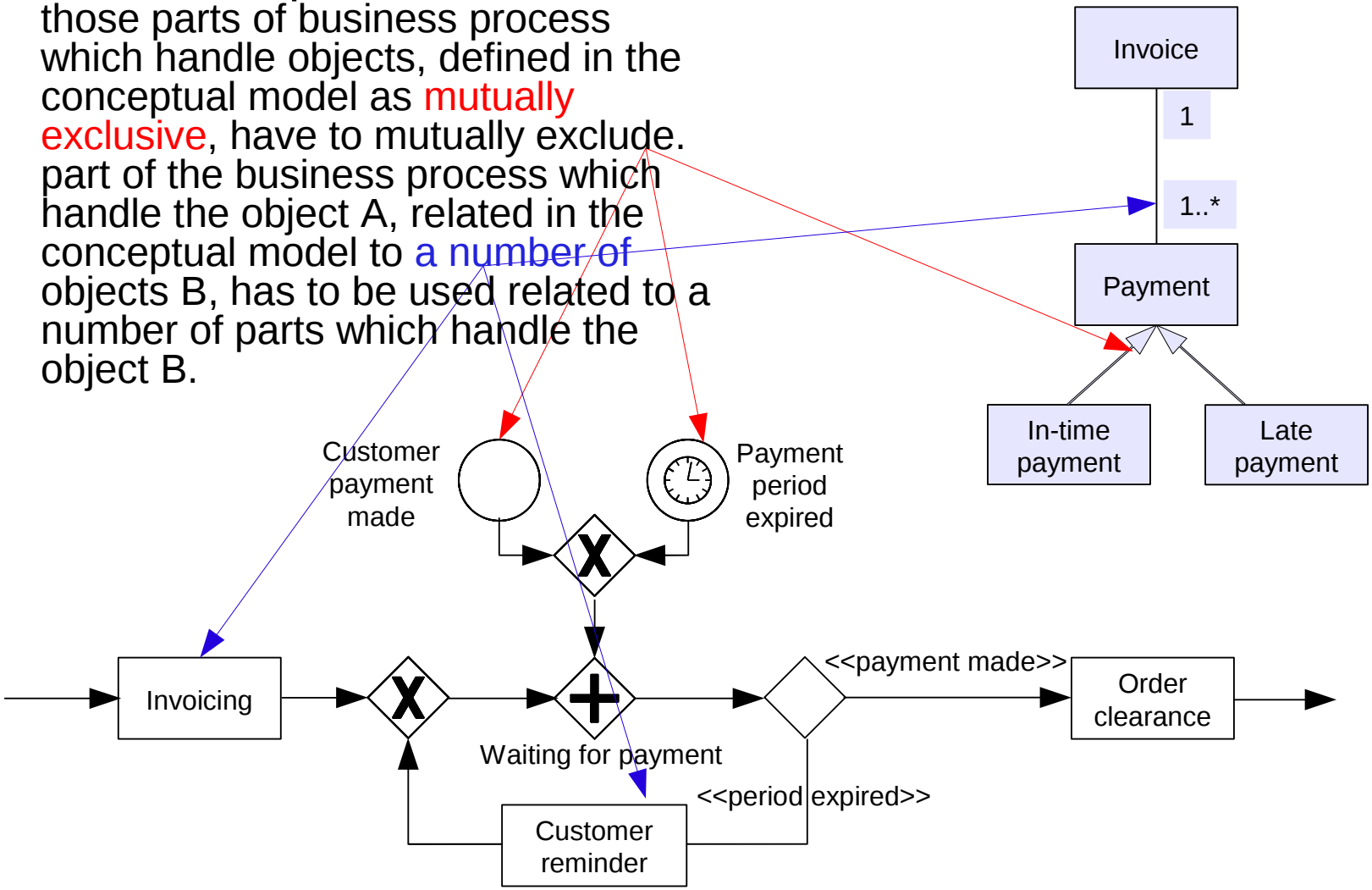
- those parts of business process which handle objects, defined in the conceptual model as mutually exclusive, have to mutually exclude.
- part of the business process which handle the object A, related in the conceptual model to a number of objects B, has to be used related to a number of parts which handle the object B.



Structural consistency

Another examples:

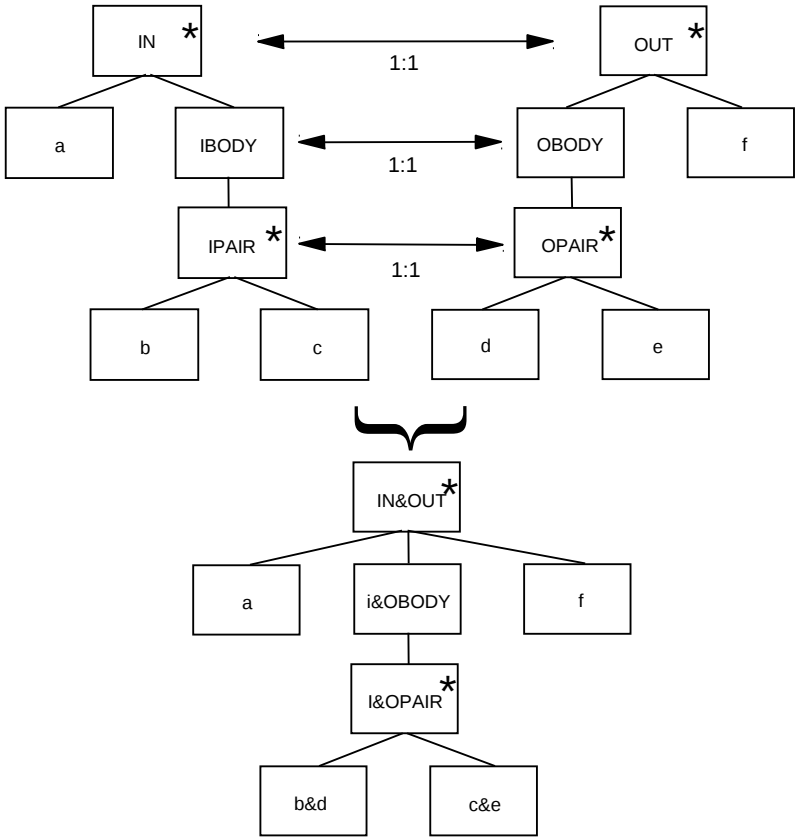
- those parts of business process which handle objects, defined in the conceptual model as mutually exclusive, have to mutually exclude.
- part of the business process which handle the object A, related in the conceptual model to a number of objects B, has to be used related to a number of parts which handle the object B.



Structural coherency by Jackson's theory

JSP - Jackson, M.A. "Principles of Program Design", Academic Press, London, 1975.
JSD - Jackson, M.A.: System Development, Prentice-Hall Inc., Englewood Cliffs, NJ, 1982.

program structure should be dictated by the structure of its input and output data streams



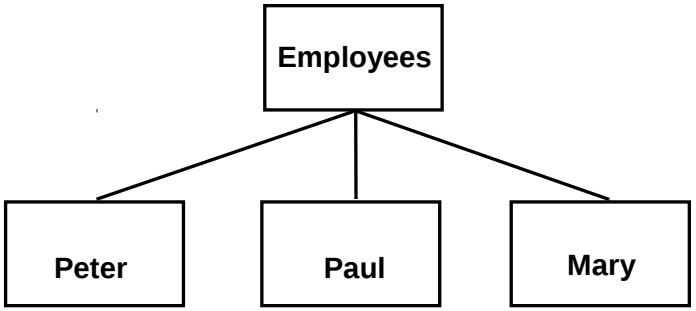
The root of the problem is solved by merging data structures.

– it requires making the set of crucial decision about the correspondences of particular data structures parts and their merging into the resulting structure (which is, in fact, the structure of transformation process from the input structure(s) to the output one(s)).

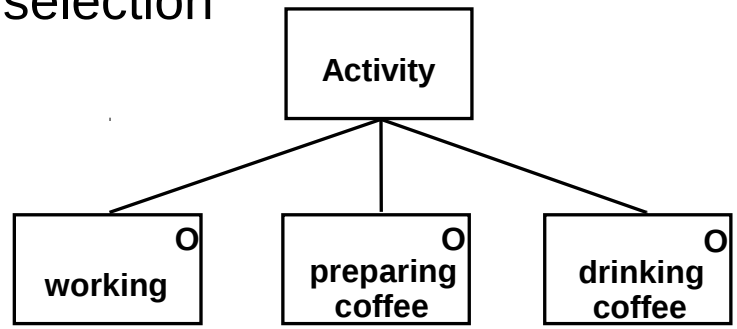
Language of Jackson's theory

„Any structure can be expressed as a hierarchy od elements using just three basic types of structure".

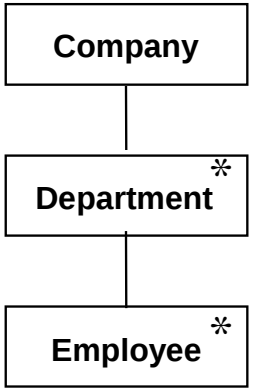
sequence



selection

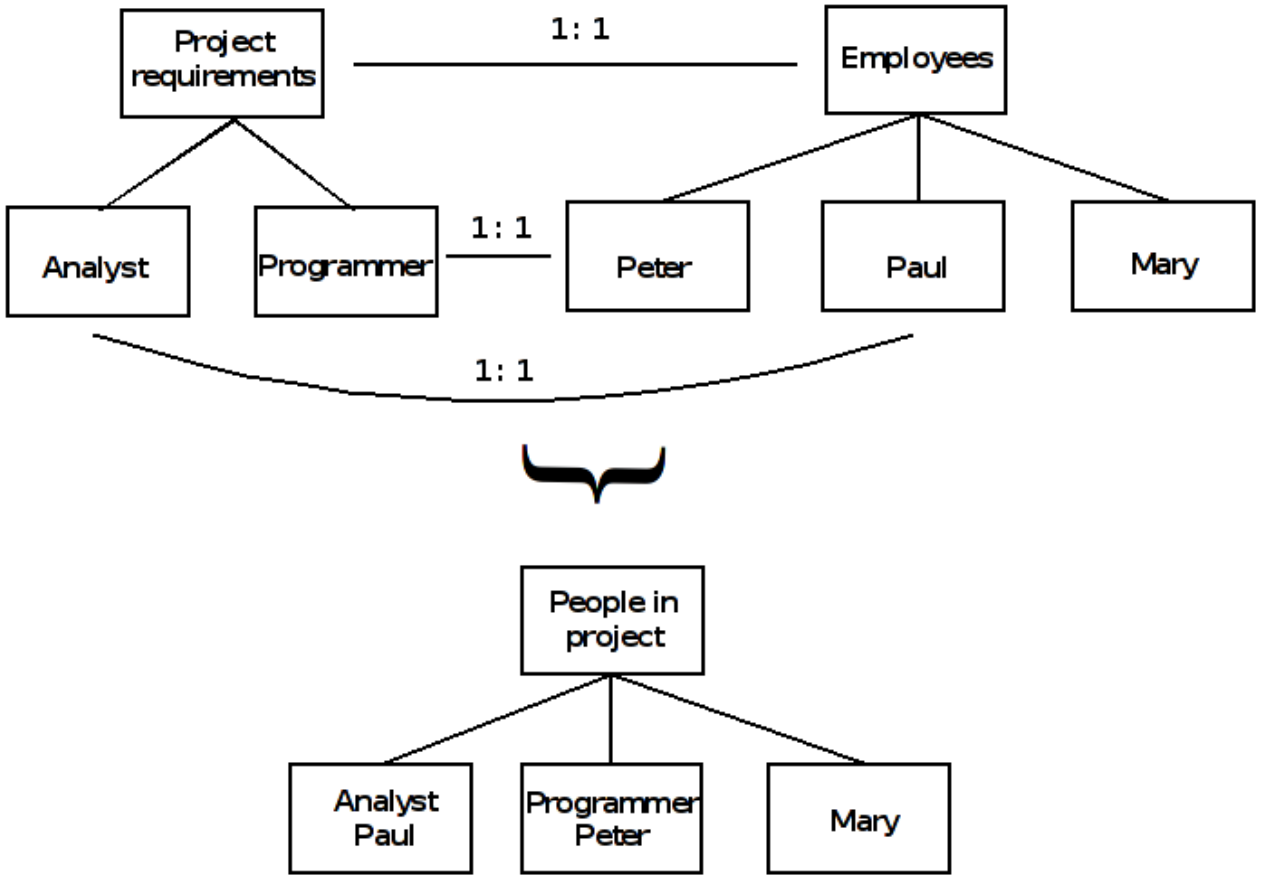


iteration



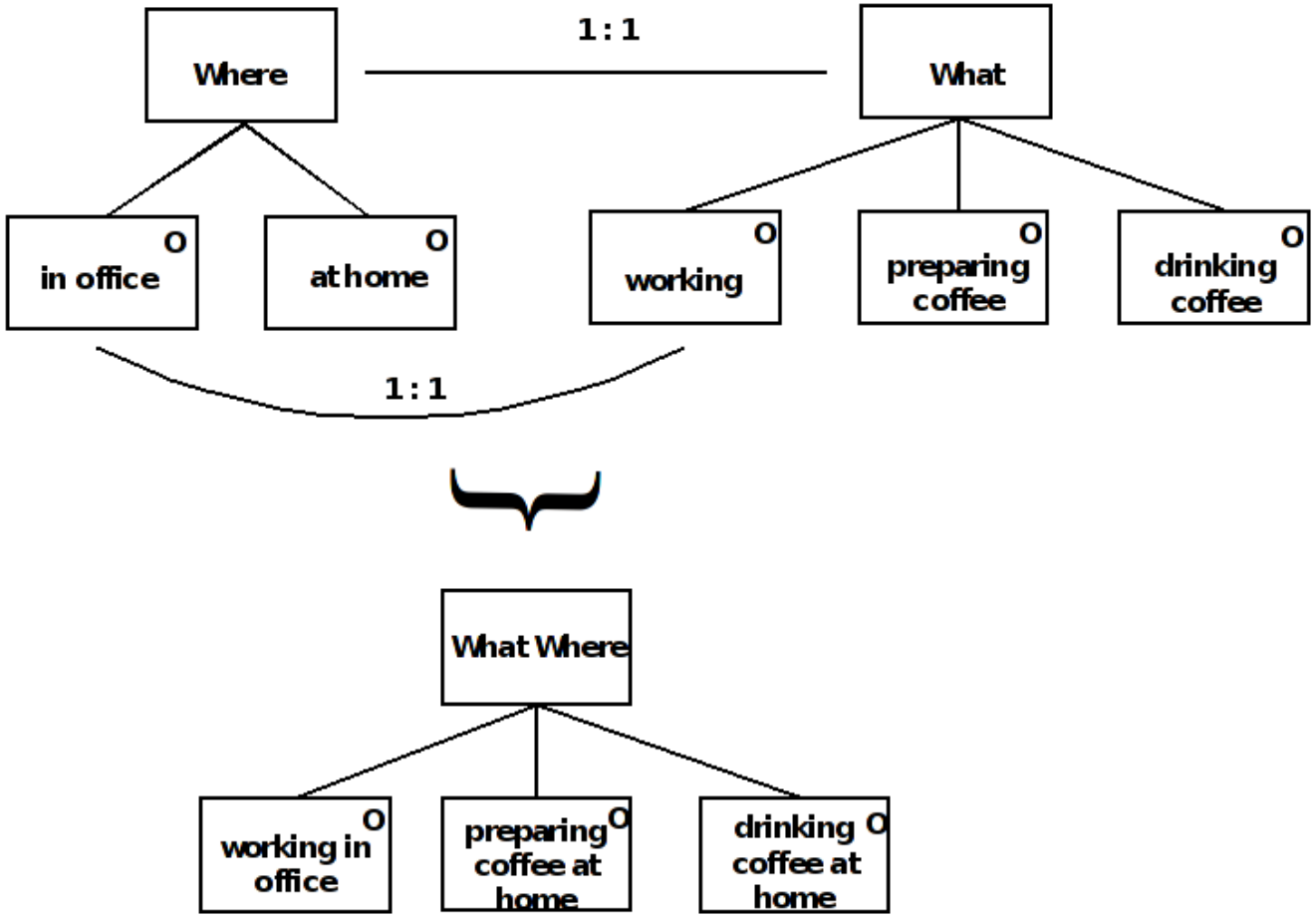
Structural coherency by Jackson's theory merging structures

sequences



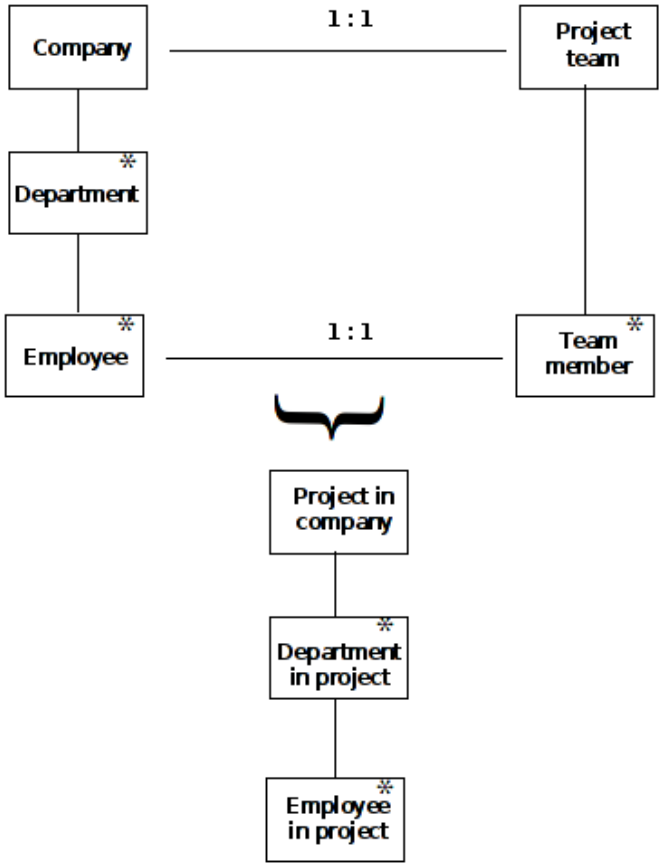
Structural coherency by Jackson's theory merging structures

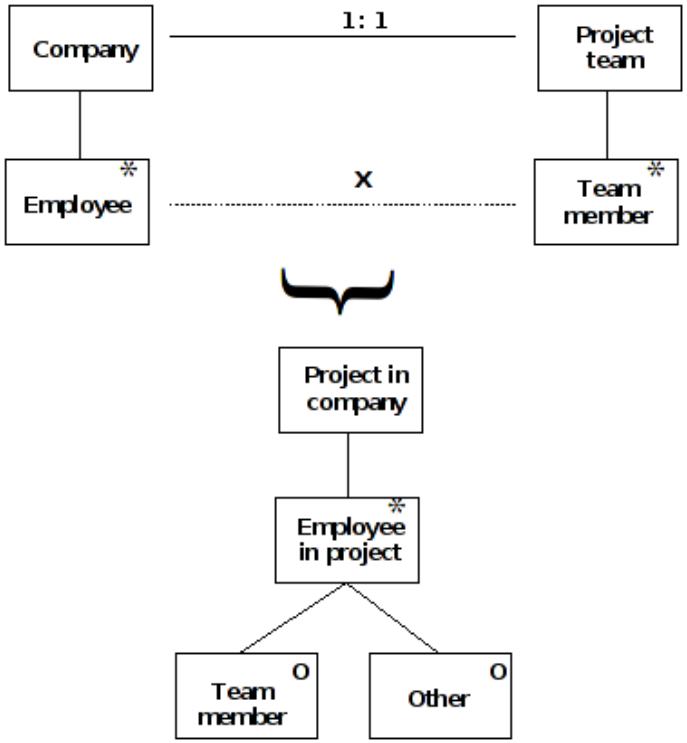
selections



Structural coherency by Jackson's theory merging structures

iterations





Mutual consistency of objects life cycles
Initial general analogies

Structural Consequences - use of Jackson's ideas[♣] for reflecting the natural consequences in Real World models - following from the ***nature of the relationships among the Real World objects***.

General analogies:

- sequence type of structure → aggregation,
- selection type of structure → generalization.
- iteration type of structure → aggregation (special case of the sequence where all its parts are of the same structure,

- cardinality of the relationships → aggregation (it reflects the quantity and says nothing about the quality),
- optionality of the relationship → generalization (it reflects the quality and says nothing about the quantity (including the ordering)).

- generalization (inheritance) should be reflected in some kind of selection,
- aggregation (composition) should be reflected in some kind of sequence / iteration, with all consequences following from it.

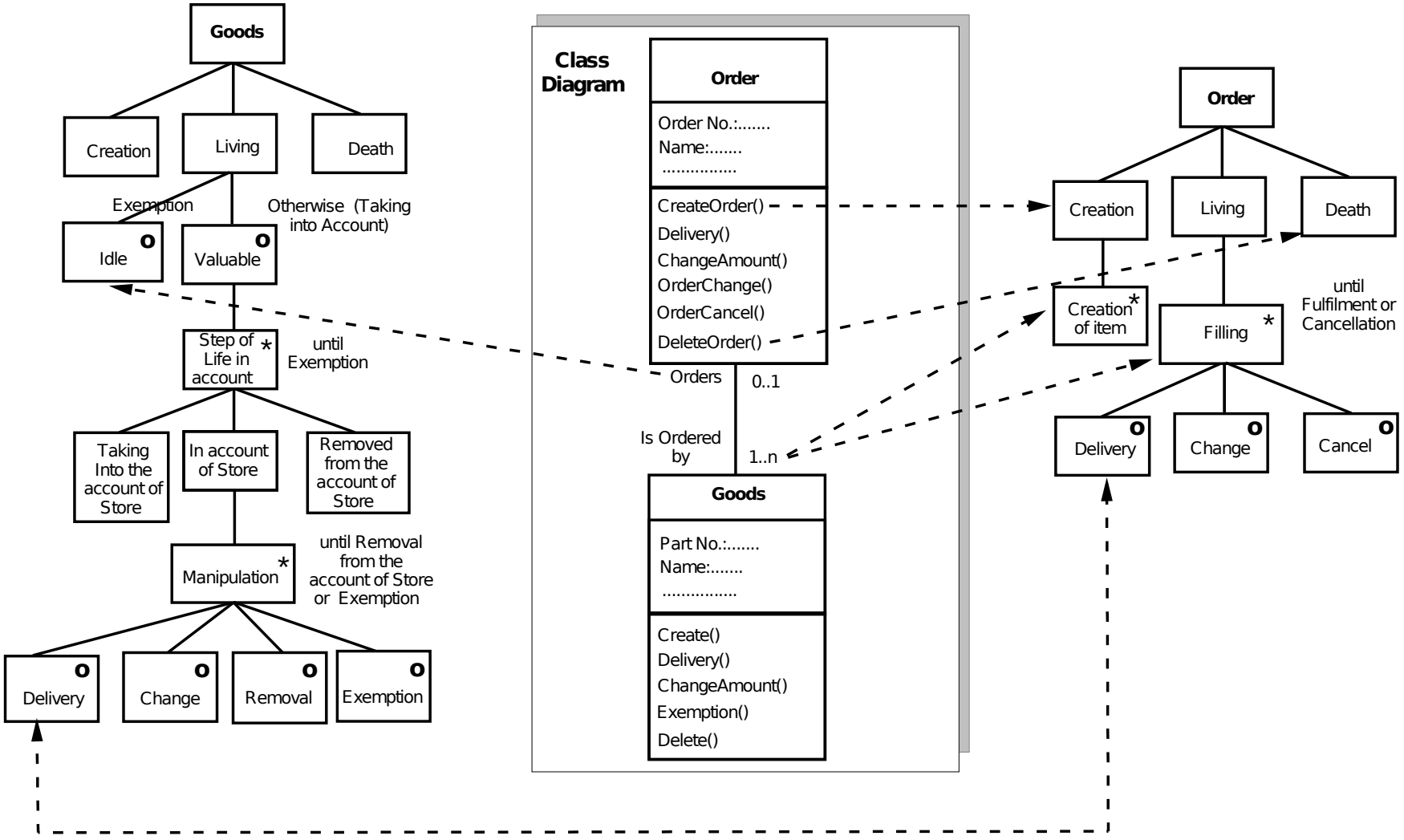
[♣] M.A.Jackson: Principles of Program Design (1975), Systems Development (1982)

Mutual consistency of objects life cycles - structural consistency of objects

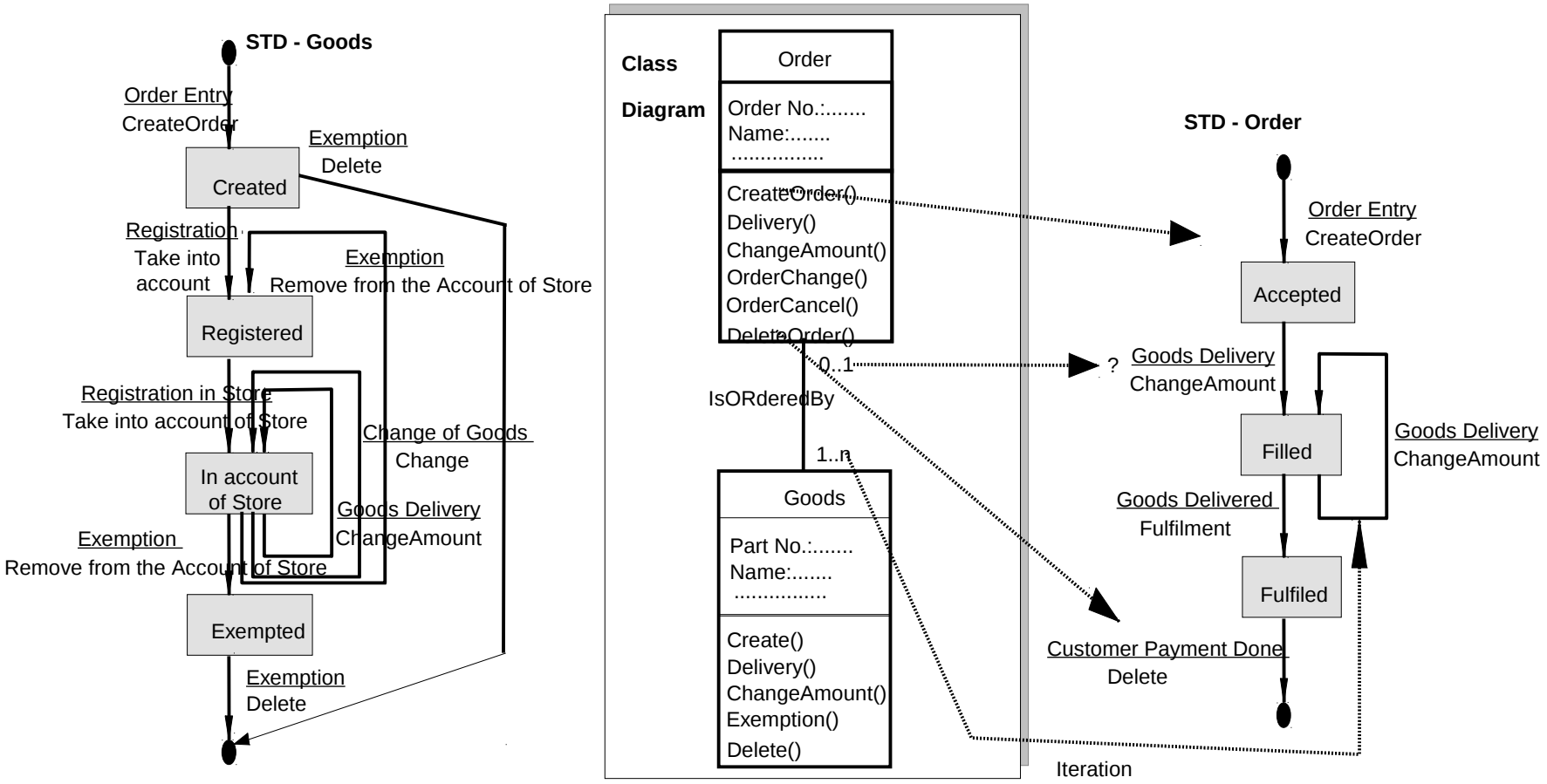
Structural consistency rules:

1. Each association between two object classes must be reflected by the specific operation(s) in each class life cycle.
2. The cardinality of the association must be reflected by corresponding type of structure in the life cycle of the opposite class:
   - cardinality 1:n by the iteration of parts,
   - cardinality 1:1 by the single part of the structure.
3. The optionality of the association must be reflected by corresponding selection structure in the life cycle of the opposite class.
4. Each generalization of the class must be reflected by corresponding selection structure in its life cycle.
5. Each aggregation association between classes must be reflected by corresponding iteration structure in the life cycle of the aggregating class (container / composite class).

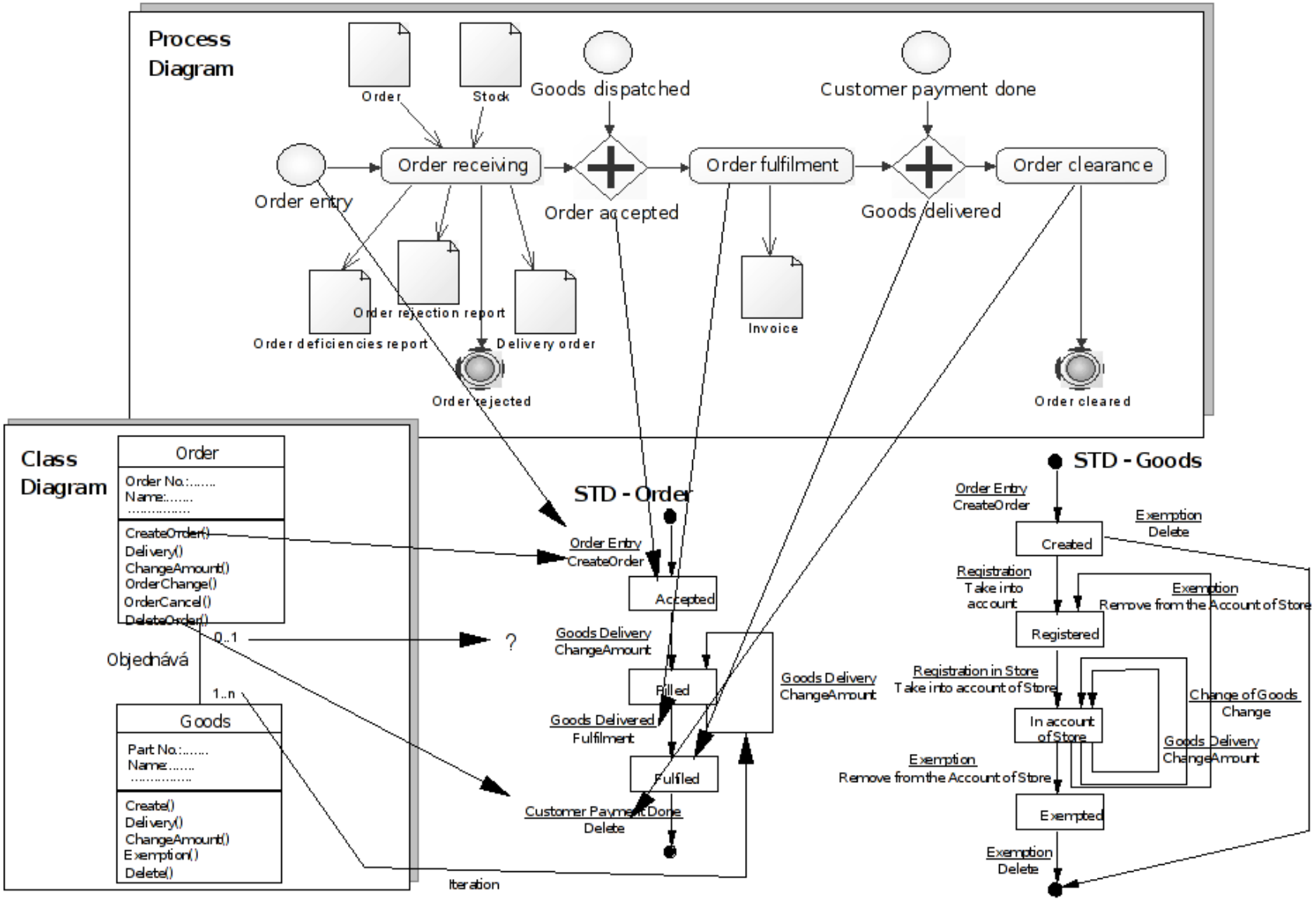
Mutual consistency of objects life cycles Example - Jackson's Diagram



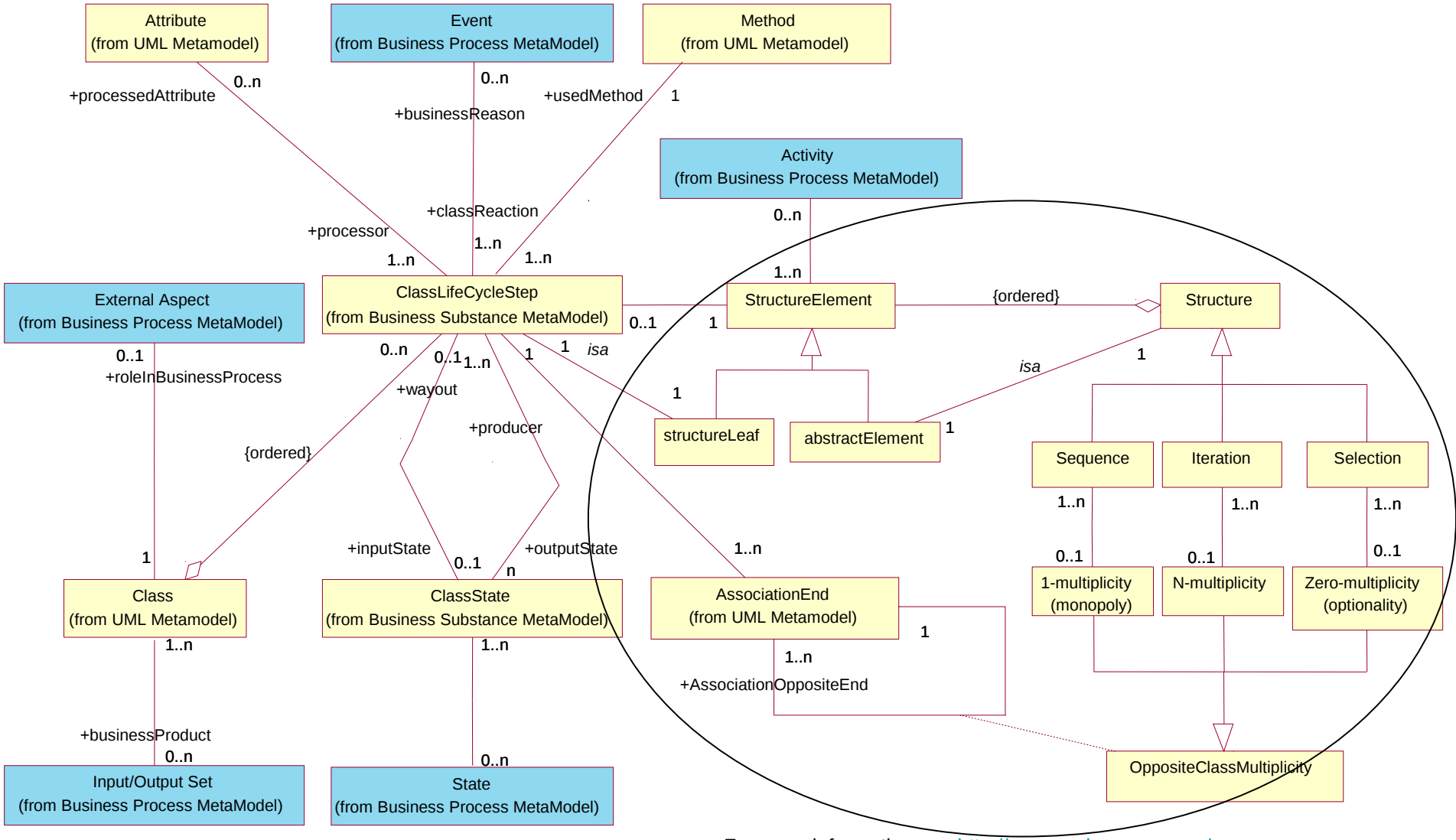
Mutual consistency of objects life cycles Example - UML



Relations among the Process and Object Models



Mutual consistency of objects life cycles
Business system meta-model (consistency package)



For more information see http://opensoul.panrepa.org/.

4. Summary

Summary

- There is the idea of **Models Coherency Criteria** in terms of **Completeness** and **Correctness** of Models.
- **Structural Consistency** is a specific type of consistency based on the ideas of M.Jackson about the natural correspondence of basic types of structures.

The concept of Consistency represents
not only a ***danger*** of inconsistency
but
an ***opportunity*** to improve the quality of models as well.

There is still a room for improving / detailing the set of Consistency Rules

The Last Word

- Presented work is the part of the project *OpenSoul*[1].
  The project is aimed on the development of the Business System Modeling Methodology based on the formal meta-model.

  For more information see
  http://opensoul.panrepa.org/metamodel.html/.

[1] *Not the source code of particular software products, but the "soul" of the software (i.e. methodology) is open in this project.*

The End