Advanced Topics in Software Construction

Winter Semester 2009/10

Final Exam (10.02.2010)

Please read the exercises carefully.
Total time: **90 min.**

Name	
Matriculation Number	

1	2	3	4	5	∑	Grade
12	14	14	12	13	65	

Remark on multiple choice tasks

Correctly marked statements result in 1 point each, incorrectly marked statements result in -1 point each. Statements left without a mark are not considered in the evaluation. The minimum overall sum you can achieve for each sub tasks is 0.

Task 1: Right or Wrong? (12 points)

a) Software Components *[4 pts]*

Right	***Wrong***	
☑	□	The two phase commit (2PC) protocol ensures data consistency when remote-procedure calls span more than two databases.
□	☑	Stateful session beans should be used with care, because they block the main memory until they are explicitly removed.
□	☑	Component frameworks must be platform and language independent.
☑	□	A stub realizes the communication between a client and a server component.

b) Service-oriented Architecture *[4 pts]*

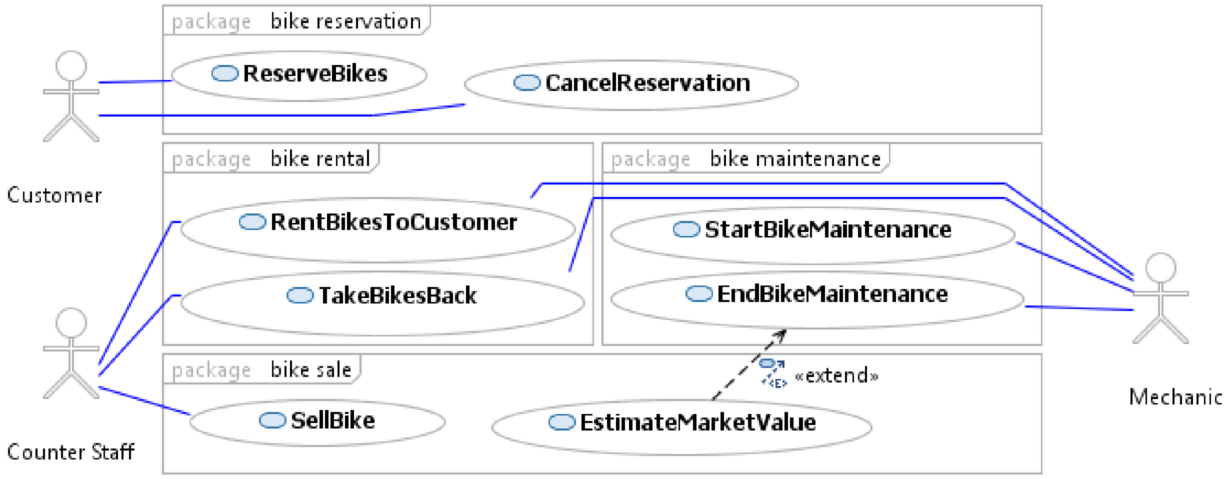
Right	***Wrong***	
□	☑	API-driven Enterprise Services Buses use proprietary protocols in contrast to protocol driven ESBs.
☑	□	Some SOAP encodings do not include the called method in the message payload.
☑	□	WSDL service specifications bind web services to concrete URLs.
☑	□	ESBs can be responsible for data-type conversion.

c) Software Processes *[4 pts]*

Right	***Wrong***	
☑	□	An iteration in the spiral model identifies and prioritizes risks and develops a series of prototypes for the risks starting with most critical.
□	☑	*Iterative* development implements and releases small functional *increments* on every iteration.
☑	□	Extreme programming suggests splitting a user story if the estimated effort exceeds a certain limit to allow for short cycle times.
□	☑	Sophisticated code structures for possible later extensions are not considered waste in lean software development.

Task 2: Domain Modeling with Archetypes (14 points)

You are developing software for different independent Bike Rental Outlets. The use case diagram should give you an idea about the required functionality. Study it!

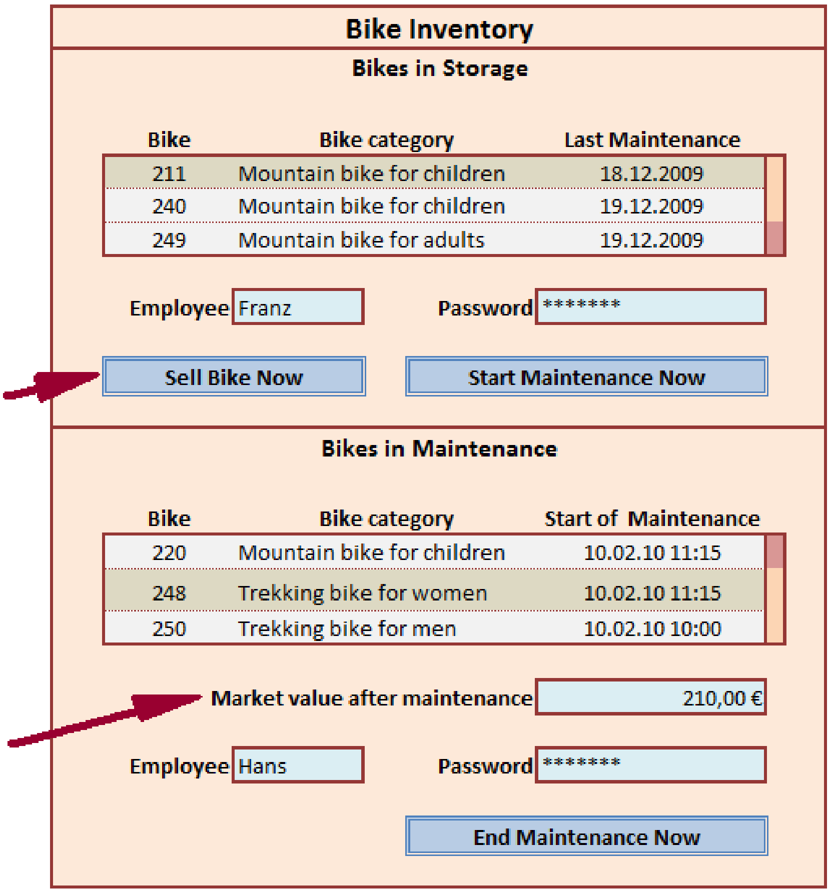


Only some outlets offer to sell bikes, all other don’t need the use cases of the package “bike sale”. In all outlets we have to expect that the same employee may act as “Counter Staff” as well as “Mechanic”. The system shall offer the following functionality:

- Reservations, Rentals, Maintenance, and Sales can be tracked with the software.
- Customers can reserve bikes online and cancel their reservation again.
- The Counter Staff can rent bikes and take them back.
  In both cases the bikes are quickly checked by a Mechanic.
- The Mechanics maintain the bikes in regular intervals.
- *Only for some outlets:* Counter staff can sell bikes. In this case Mechanics always re-estimate the market value of the maintained bike after the maintenance.

On the right you see the bike inventory form that is used by the mechanics and counter staff for the maintenance, estimation and sale use case.

The two arrows indicate the elements that are only available for some outlets. (You will discuss this in task 3.)



Bike Inventory

Bikes in Storage

Bike	Bike category	Last Maintenance
211	Mountain bike for children	18.12.2009
240	Mountain bike for children	19.12.2009
249	Mountain bike for adults	19.12.2009

Employee Franz Password *******

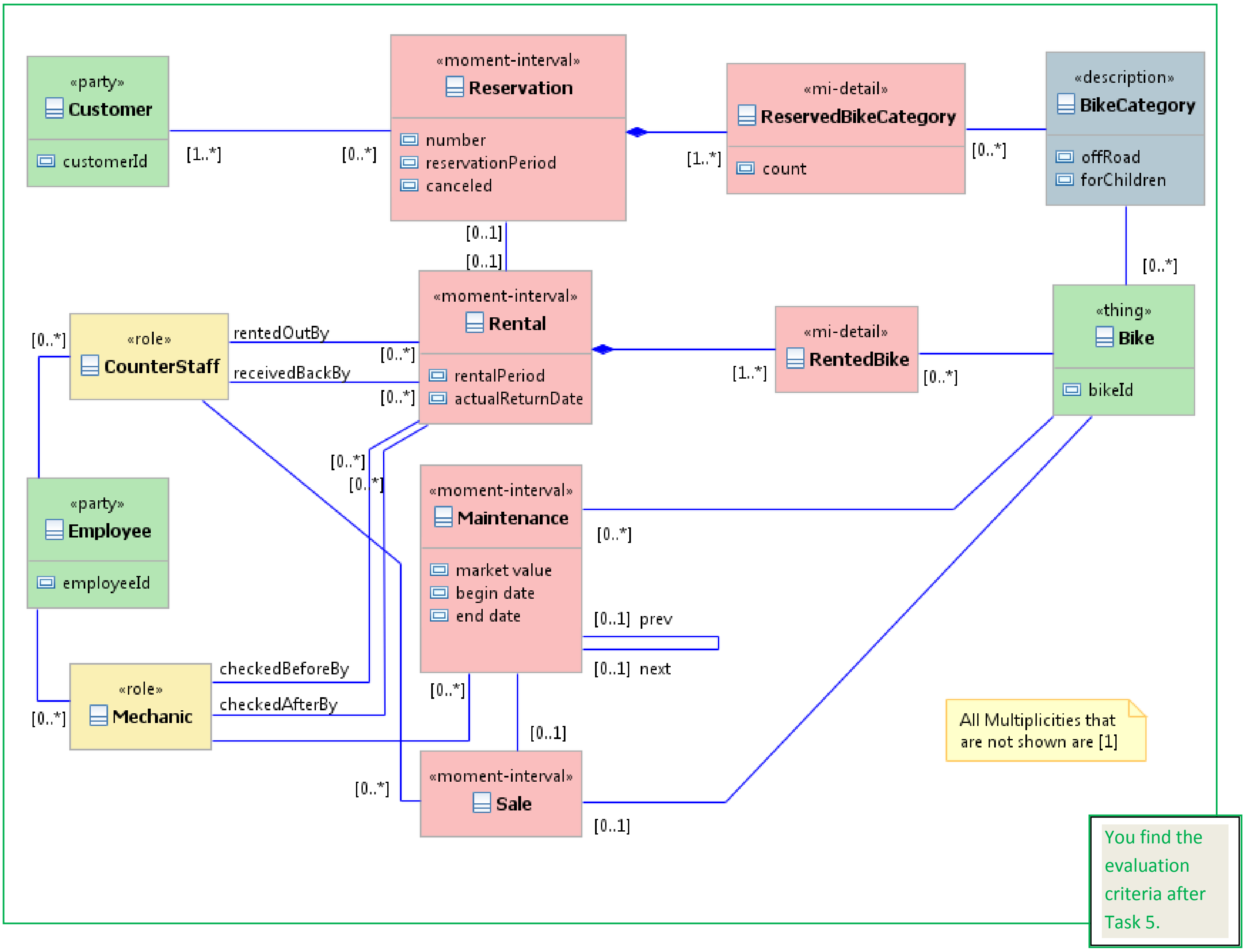
Bikes in Maintenance

Bike	Bike category	Start of Maintenance
220	Mountain bike for children	10.02.10 11:15
248	Trekking bike for women	10.02.10 11:15
250	Trekking bike for men	10.02.10 10:00

Market value after maintenance 210,00 €

Employee Hans Password *******

a) ***Create the Domain Object Model for the software using the archetypes introduced by Peter Coad (Modelling in Color). Add Multiplicities!*** ***[10 pts]***

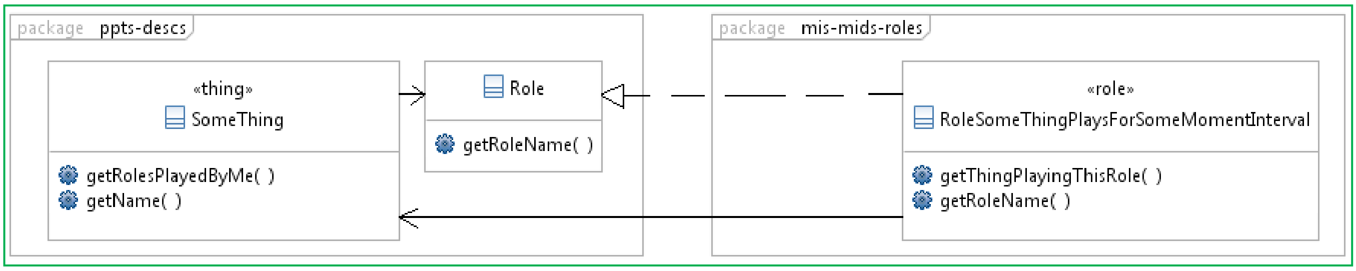


You find the evaluation criteria after Task 5.

b) ***One strategy to organize colored models is to build packages of related «description», «party», «place», and «thing»s on the one hand, and packages of related «role», «moment-interval», «mi-detail»s on the other hand.***
Would you prefer dependencies from «description»-«party»-«place»-«thing» packages to «role»-«moment-interval»-«mi-detail» packages or in the other direction? Why? ***[2 pts]***

Dependencies on more stable and more abstract packages are preferable. Things etc. are more abstract because they can be used for some potential applications. Moment-Intervals are less stable as they change with every new business process. Consequently the "moment-interval, ..." package should depend on the "thing, ..." package.

c) ***(Tricky!) «thing»s typically offer a method returning the «role»s the «thing» is playing. Concrete «role»s reference the «thing» that plays the role. How can we package both in separate packages as suggested in b) but avoid introducing dependencies in both directions? (Tip: Introduce an extra type)*** ***[2 pts]***



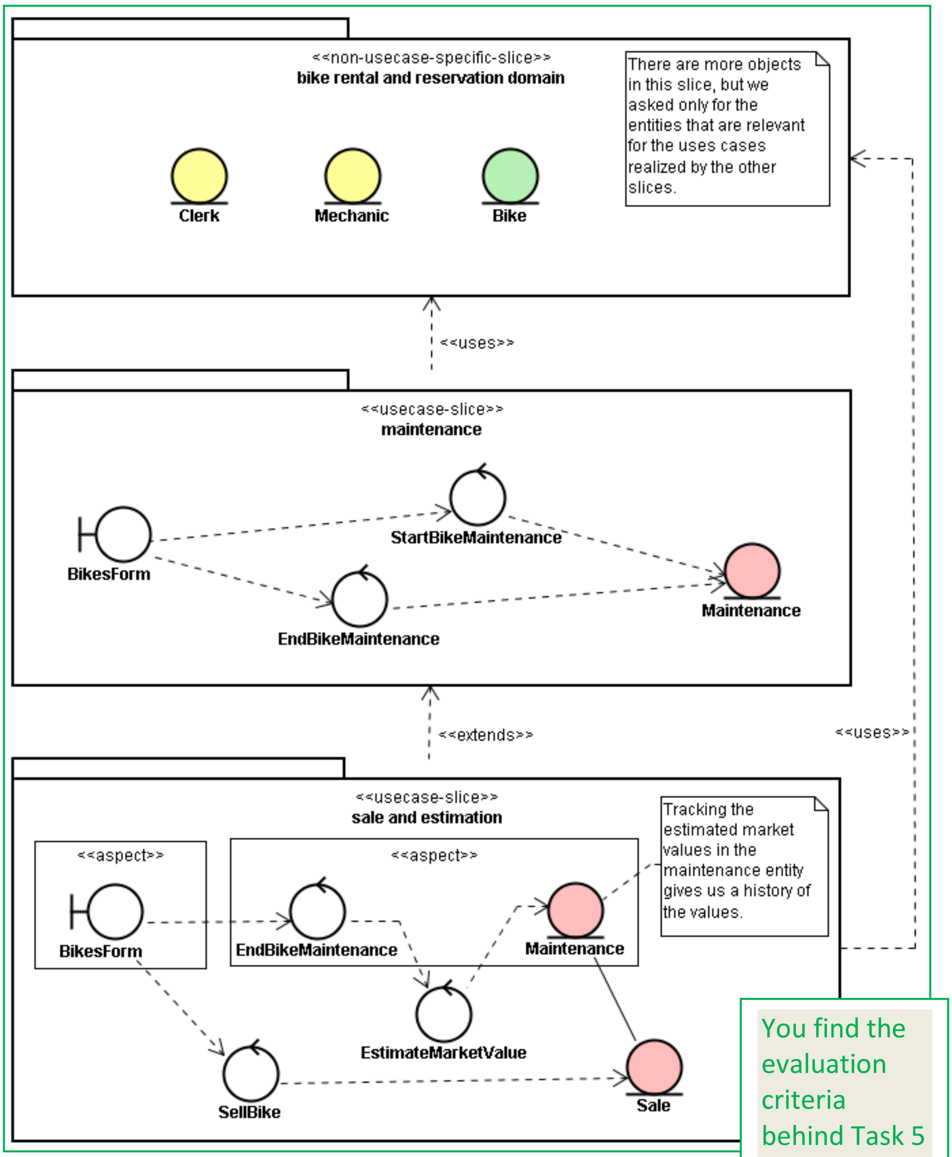
Task 3: AOSD with Use Cases (14 points)

Make sure that you read task 2. The diagram on the right shows a part of the analysis object model for our application. We created the analysis objects that are necessary to realize the maintenance use cases.

a) ***Add to the «non-usecase-specific-slice» those entities that are required by the maintenance use cases. [1 pt]***

b) ***Create the analysis objects that are necessary to realize the "Sale Bike" and "Estimate Market Value" use cases in the «usecase-slice» "sale and estimation". [8 pts]***

Use the aspect-oriented approach where useful. Don't forget to give us the dependencies between the slices. If necessary add domain objects to the «non-usecase-specific-slice».



<<non-usecase-specific-slice>>
bike rental and reservation domain
Clerk
Mechanic
Bike
There are more objects in this slice, but we asked only for the entities that are relevant for the uses cases realized by the other slices.
<<uses>>
<<usecase-slice>>
maintenance
BikesForm
StartBikeMaintenance
EndBikeMaintenance
Maintenance
<<extends>>
<<uses>>
<<usecase-slice>>
sale and estimation
<<aspect>>
BikesForm
<<aspect>>
EndBikeMaintenance
Maintenance
Tracking the estimated market values in the maintenance entity gives us a history of the values.
EstimateMarketValue
SellBike
Sale

You find the evaluation criteria behind Task 5

c) ***Explain the semantics of an «include» relationship between two use cases. Which relationship do you expect between the use case-slices that realize them? [2 pts]***

Use case A <<includes>> use case B if the flow of events of A always includes the events a B at a certain point of the flow. We use <<uses>> to model this relationship on the level of use case slices.

d) ***We used an «extends» relationship for the relation between the use cases "End Bike Maintenance" and "Estimate Market Value". Do you think this was an appropriate choice? Why? (No points for the yes/no answer but for the reason) Which relationship do you expect between the use case-slices that realize them? [3 pts]***

YES, because the <<extends>> relationship means that the events of the extending behavior are only sometimes (optional or on exceptional cases) part of the flow of the extended use case, and the estimation of the market value is not for all shops part of the end maintenance use case.

[Also ok: NO, because the being optional in use case diagrams means being optional during the execution of the extended use case. Here the "extending" use case is only optional at deployment time.]

We use <<extend>> in this case between the use case slices.

Task 4: Requirements Improvement (12 points)

One of the Bike Rental Outlets wants to make an additional service available to their customers. They know that cycling is more fun with a guide. You get a note from the manager of the outlet with some preliminary requirements. From a quick view on the text you already know that some information he had in mind didn't make it to the written note (was "deleted"), some statements are more general than necessary, and some are simply distorted. Prepare for improving these "requirements".

> *(1) The reservation may include a tour guide reservation request. (2) The reservation will be confirmed. (3) At least three of the participating bikes must be rented from the shop. (4) If the trip is longer, customers will be charged for a full day. (5) If necessary, the guide can help you to make use of the tools.*

a) Review these "requirements" using the Sophist REgelwerk! [8 pts]

Give us <u>five</u> *issues* of *at least* <u>four</u> *different types*. You should use what you learned about the Sophist REgelwerk and the other rules.

Sen-tence	Clue on the language level pointing to a problem?	What is the problem?	What could you ask to clarify the requirement?
1	Modal verb "may"	Conditions not clear	What are the prerequisites to include a tour guide reservation request in a reservation?
2	Passive voice	Who/What confirms the reservation?	Who/What confirms the reservation?
4	Comparison "longer"	No compared object in the "comparison"	How long must the trip be in order to charge for a full day?
5	"If necessary"	"necessary" not specified	Under which conditions should the guide help to make use of the tools?
1	"tour guide reservation request"	Nominalization might lead to distortion	Who creates the reservation request? How does the request relate to an actual reservation?

b) Give us <u>two</u> examples how a rule about how to phrase your sentences in natural language can prevent these problems. [4 pts]

Describe each rule in your own words.
Demonstrate how it prevents the problem using a *sentence* e.g. from the note above.

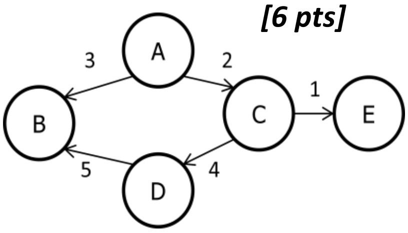
Definitions of nouns should have the form:
<<noun>> is a <<more general noun>> <<description>>
Example: "A *tour guide reservation request* is a request to reserve a guide for a bike tour."
(Reveals new problems)

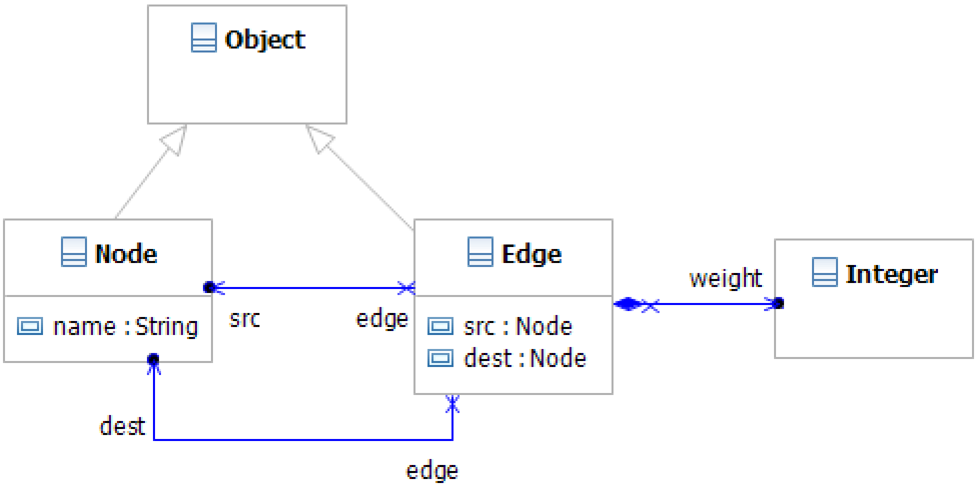
Active voice should be used.
Example: "The tour guide confirms the reservation"
(Clarifies responsibilities)

Task 5: Model-driven Software Development (13 points)

a) Create a MOF-based Meta-Model for the following domain ***[6 pts]***

Assume we want to model directed graphs consisting of **nodes** and weighted, directed **edges**. Each node has a name and each edge has a weight of type Integer assigned.

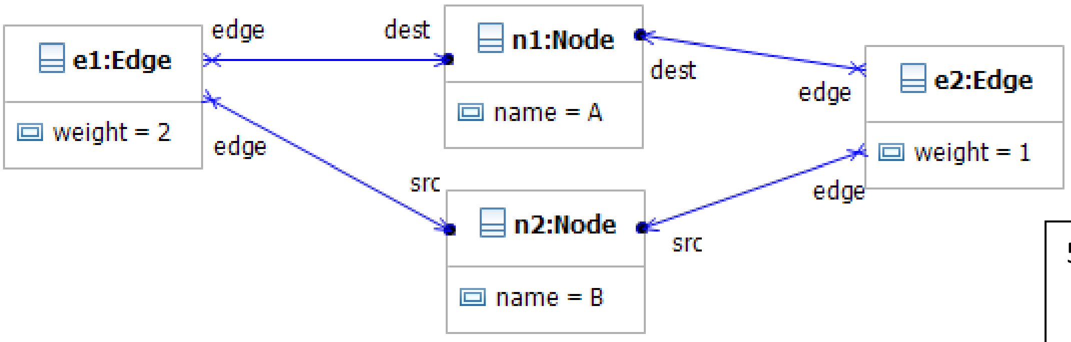




5a) Correct	
Classes	2P
Attributes	2P
Aggregation	1P
Role names	1P

b) Instantiate the meta-model with an object-diagram ***[3 pts]***

The model should contain at least three nodes and two edges.



5b) Correct	
Syntax/Sem.	1P
Objects	1P
Attributes	1P

c) What are the main advantages and disadvantages of UML Profiles in comparison to a MOF-based Meta-model? ***[4 pts]***

UML Profile Advantages

- **Broad tool support available** **1P**
- **Rich Meta-Model for OO-Modeling** **1P**
- **Easy extensible** **1P**

MOF-based Meta-Model Advantages

- **Tailored to the needs of the concrete domain** **1P**
- **Suitable for Non-OO Modeling** **1P**

Evaluation Criteria

Task 2: Domain Modeling with Archetypes (14 points)

Evaluation of 3a)

- 2 Points: Correct Classes found (Counter Staff, Mechanic, Maintenance, Sale, Bike, market value; 0,5 if at least one was found; 1 if three and 1,5 if 4 found; 2 if five or six found)
- 6 Points: Modeling in Color
  - 1 Point: Bike Category found and connected to Bike
  - 1 Point: Roles found and used correctly
  - 2 Points: Appropriate selection of stereotypes
  - 2 Points: Consistency following the Color Modeling approach
- 2 Points: Correct Multiplicities

Task 3: AOSD with Use Cases (14 points)

Evaluation 3b)

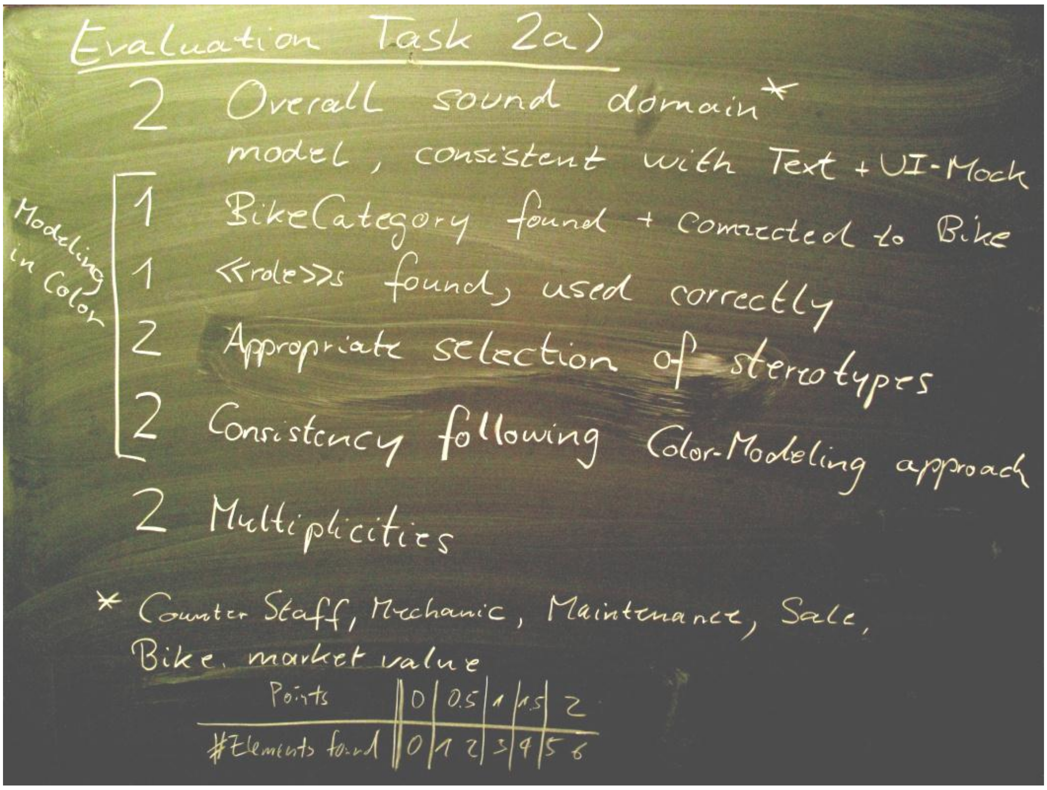
- 3 Points for AOSD
  - Understanding of use-case-slice diagrams; Slice <<extends>> other slice if aspect applies on other slice; Objects at most once outside an aspect; Meaningful <<aspect>> boxes.
- 3 Points for AOP
  - Extending the right objects with the <<aspect>> mechanism
- 3 Points for OOP
  - Sound analysis object model

Task 4: Requirements Improvement (12 points)

Evaluation 4a)

- 4 Points: Identification of flaws (First column of our dashes)
- 4 Points: Application of a Sophist Rule (Second column of our dashes)

We count the four best rows as long as they contain at least three different kinds of problems.



Evaluation Task 2a)

2 Overall sound domain*
model, consistent with Text + UI-Mock

Modeling in Color

1 BikeCategory found + connected to Bike

1 <<role>>s found, used correctly

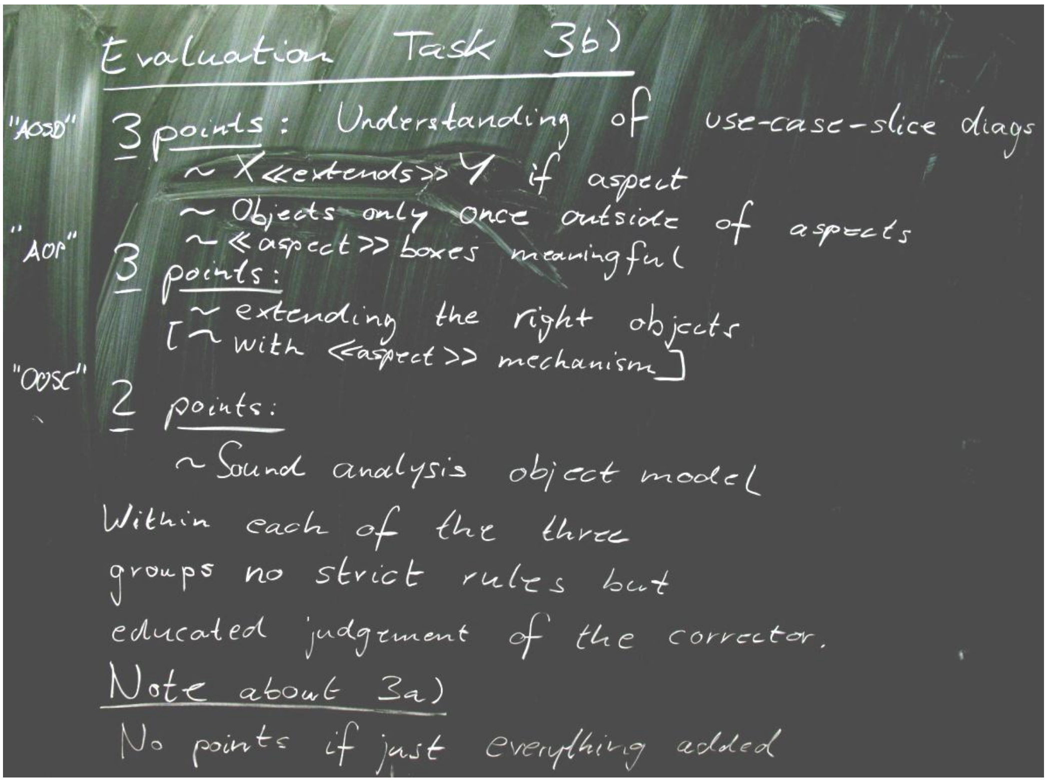
2 Appropriate selection of stereotypes

2 Consistency following Color-Modeling approach

2 Multiplicities

* Counter Staff, Mechanic, Maintenance, Sale,
Bike, market value

Points	0	0.5	1	1.5	2
#Elements found	0	1 2	3	4	5 6



Evaluation Task 3b)

"AOSD" 3 points: Understanding of use-case-slice diags
~ X <<extends>> Y if aspect
~ Objects only once outside of aspects
~ <<aspect>> boxes meaningful

"AOP" 3 points:
~ extending the right objects
[~ with <<aspect>> mechanism]

"OOSC" 2 points:
~ Sound analysis object model

Within each of the three
groups no strict rules but
educated judgement of the corrector.

Note about 3a)

No points if just everything added