

Assignment Sheet 4

Due: Tuesday, November 15, 2011, 23:59

For help, contact atsc-lecture@lists.iai.uni-bonn.de (staff and participants).

Start working on the exercises early enough so that you can contact the staff in time in case of problems. Don't expect them to be available during the weekend!

Submit your solution into your group's SVN under the folder "assignment04".
The solutions (if not otherwise mentioned) must be submitted as **PDF files**.

Task 1. *Domain Model* (3 Points)

The two dialogs (A and B) on the next page have subtle, but important differences. Answer for each dialog the following questions with 1-2 sentences and explain your opinion:

- a.) (1 Point) Are the user and the developer speaking the same language?
- b.) (1 Point) Is that language rich enough to carry the discussion of what the application must do?
- c.) (0.5 Points) What are the differences of a.) and b.) between the two dialogs?
- d.) (0.5 Points) Which of the two dialogs would you prefer? Explain your answer.

Task 2. *Identifying Entities and Value Objects* (4 Points)

One of the core questions in domain modeling is, whether a domain element is an entity or a value object. This answer mostly depends on your domain. Present for each element a domain in which it represents an entity and one domain in which it represents a value object. Prefer a short description for your argumentation in one or two sentences.

- a.) Country
- b.) Address
- c.) Scientific Publication
- d.) Computer Hardware

Task 1 Dialog A

User: So when we change the customs clearance point, we need to redo the whole routing plan.

Developer: Right. We'll delete all the rows in the shipment table with that cargo id, then we'll pass the origin, destination, and the new customs clearance point into the Routing Service, and it will repopulate the table. We'll have to have a Boolean in the Cargo so we'll know there is data in the shipment table.

User: Delete the rows? OK, whatever. Anyway, if we didn't have a customs clearance point at all before, we'll have to do the same thing.

Developer: Sure, anytime you change the origin, destination, or customs clearance point, we'll check to see if we have shipment data and then we'll delete it and then let the Routing Service regenerate it.

User: Of course, if the old customs clearance just happened to be the right one, we wouldn't want to do that.

Developer: Oh, no problem. It's easier to just make the Routing Service redo the loads and unloads every time.

User: Yes, but it's extra work for us to make all the supporting plans for a new itinerary, so we don't want to reroute unless the change necessitates it.

Developer: Well, then, if you are entering a customs clearance point for the first time, we'll have to query the table to find the old derived customs clearance point, and then compare it to the new one. Then we'll know if we need to redo it.

User: You won't have to worry about this on origin or destination since he itinerary would always change then.

Task 1 Dialog B

User: So when we change the customs clearance point, we need to redo the whole routing plan.

Developer: Right. When you change any of the attributes in the Route Specification, we'll delete the old Itinerary and ask the Routing Service to generate a new one based on the new Route Specification.

User: If we hadn't specified a customs clearance point at all before, we'll have to do that at the same time.

Developer: Sure, anytime you change anything in the Route Spec, we'll regenerate the Itinerary. That includes entering something for the first time.

User: Of course, if the old customs clearance just happened to be the right one, we wouldn't want to do that.

Developer: Oh, no problem. It's easier to just make the Routing Service redo the Itinerary every time.

User: Yes, but it's extra work for us to make all the supporting plans for a new Itinerary, so we don't want to reroute unless the change necessitates it.

Developer: Oh. Then we'll have to add some functionality to the Route Specification. Then, whenever you change anything in the Spec, we'll see if the Itinerary still satisfies the Specification. If it doesn't we'll have the Routing Service regenerate the Itinerary.

User: You won't have to worry about this on origin or destination since the Itinerary would always change then.

Task 3. *Intention revealing names* (3 Points)

A program for paint stores can show a customer the result of mixing standard paints. The initial design contains only a single domain class. Further, the functionality of the class is partially described in the following code snippets.

Paint
v : double
r : int
y : int
b : int
paint(Paint)

Code Fragment:

```
public void paint(Paint paint){
    v = v + paint.getV(); // after mixing, volume is summed
    // omitted many lines of complicated color mixing logic
    // ending with the assignment of new r,b, and y values.
}
```

Test case:

```
public void testPaint(){
    // create a pure yellow color with volume = 100
    Paint yellow = new Paint(100.0,0,50,0);
    // Create a pure blue color
    Paint blue = new Paint(100.0,0,0,50);
    // Mix blue into the yellow color
    yellow.paint(blue);
    // Result should be green
    assertEquals(25,yellow.getB());
    assertEquals(25,yellow.getY());
    assertEquals(0,yellow.getR());
}
```

- (2 Points) Refine the domain model to express the domain elements from the client's point of view. If needed you can add domain objects and/or rename by using intention revealing names.
- (1 Point) Explain your solution and how you derived it (1-3 sentences).

Hint: Also consider comments when searching for hidden domain elements.

Task 4. *Ubiquitous Language* (5 Points)

Imagine you are working in a company and should create a spell-checking software for a client. To distillate the domain model one of the first steps is to create an initial ubiquitous language. Your task is to create a *Concept Map* of the ubiquitous language which contains the elements and relations of the domain. Your map should contain 5-10 elements. Please submit your final concept map as an image file.

Hint: From the Wikipedia article on Concept Maps (http://en.wikipedia.org/wiki/Concept_map):
A concept map is a diagram showing the relationships among concepts. It is a graphical tool for organizing and representing knowledge. Concepts, usually represented as boxes or circles, are connected with labeled arrows in a downward-branching hierarchical structure. The relationship between concepts can be articulated in linking phrases such as "gives rise to", "results in", "is required by," or "contributes to".

Hint: A good strategy for solving this task is to do a brainstorming about the domain in your group.

Dictionary Task* (0 Points)

On each assignment sheet you will find a set of terms used in the lecture. Your (optional) task is to write a short description for each term in your own words. Try to make it as short and precise as possible. We do not accept copied sentences.

The best solution of each week will be added to a combined dictionary on our lecture website:

<http://sewiki.iai.uni-bonn.de/teaching/lectures/atsc/2011/dictionary>

There you can also see which group contributed which description. You can also at any time improve an existing entry.

Creating the dictionary is also an opportunity to check your knowledge on the topics. This also directly gives you feedback on which topics you probably need to improve for the exam.

Terms:

Entity

Context Map

Shared Kernel

Aggregate

Value Object