Component-Based Software Engineering	**Exercise Sheet No. 2**
Dipl.-Inf. Florian Heidenreich Office Hours: Wednesdays, 1430–1530 hours INF 2080 `http://st.inf.tu-dresden.de/teaching/cbse`	Software Technology Group Institute for Software and Multimedia Technology Department of Computer Science Technische Universität Dresden 01062 Dresden

Development of Component-Based Applications

Task 1: Real estate management system

A real estate management company maintains and lets various types of buildings on behalf of its clients. Each building holds a number of apartments, each of which has a certain size and price. The company has two types of customers: owners of real estate managed by the company, and tenants, who rented one (or more) of the apartments.

Design a web-based apartment management system. Concentrate on the following two use cases:

1. An apartment that is currently not let, gets rented out to a tenant. The tenant may already be in the database, but it may also be a new tenant. The tenant must sign a lease for the rented apartment.
2. The real estate company needs to provide an owner of some buildings with a balance sheet with the financial status of all real estate managed for this owner.

Following the UML Components approach of Cheesman and Daniels [1], develop a component-based design for this application.

1a)

Create a business concept model from the description above.

1b)

For each of the two use cases define a sequence of steps that need to be performed by an actor using the system to execute the use case. Who are the actors for each use case?

1c)

Derive system interfaces and their corresponding operations from the use case definitions.

1d)

Derive a business type model from your business concept model. Can you identify business rules? (There are none explicitly mentioned in the description above, but maybe you can think of some that make sense to you?)

Identify the core types and derive corresponding business interfaces. Allocate responsibility for each type from the business type model to one business interface. What do you do with associations that cross component boundaries?

Summarise your results so far through a diagram of a rough system architecture. Discuss how you allocate interfaces to components. In particular, which components implement more than one interface, which only one?

1e)

Use the use cases, your system interfaces, and your first system architecture to discover operations in the business interfaces. Use collaboration diagrams to describe how these operations are used.

1f) *

Using interface information models and Object Constraint Language constraints, precisely specify the semantics of the operations you defined.

1g)

Discuss how you could implement your system using Enterprise JavaBeans technology. In particular, discuss what entity and session beans you would create.

References

1 John Cheesman and John Daniels. *UML Components: A Simple Process for Specifying Component-Based Software.* Component Software Series. Addison-Wesley, 2001. ISBN 0-201-70851-5