

Designing Component-Based Systems

Task 1: Components

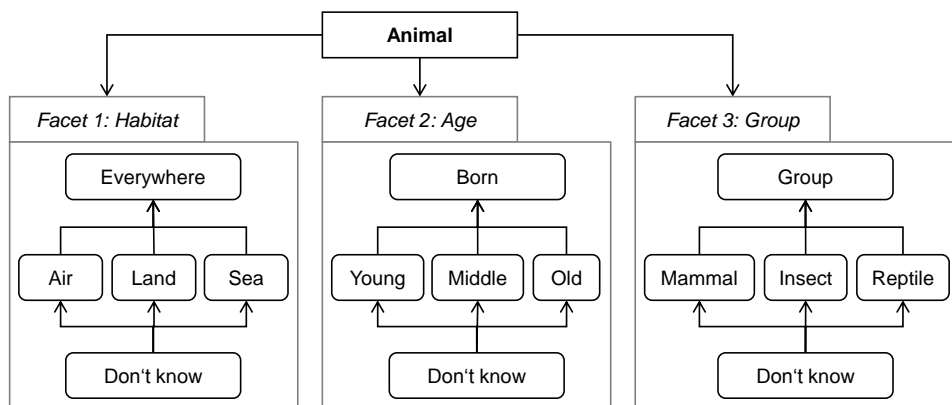
Components are the central elements of component-based systems. This task repeats the terminology and the fundamentals of components.

1a) Task: Malcolm Douglas McIlroy was talking about *Components Off The Shelf*, already in 1969 [1]. What is the main idea of COTS and why would this be beneficial?

Solution: The idea of Components off the shelf in software engineering is based on the standardized components in production. When the components comply with a common standard w.r.t. to their shape, function and interfaces, then no individualization is necessary. Furthermore different variants of different vendors are exchangeable. That way, software systems can be developed by composing existing components from various vendors.

1b) Task: What is a facet and what is facet classification? Give an example.

Solution: Facets are orthogonal (independent) dimensions of a model. Every facet represents an independent classification of an entity.



1c) Task: What is a *component repository*, a *component market* and a *component trader*?

Solution:

A Component Repository is an environment where components can be distributed and obtained.

A Component Market is a potentially commercial public component repository with a component trader.

A Component Trader is an engine for describing, finding and obtaining components. The description/search can be based on various properties (functionality, contracts, protocols etc.).

1d) Task: What are the key modeling elements of *UML Components*? What is the graphical notation? Give an example.

Solution: A UML-Component has a name, provided and required ports (interfaces) and may have more internal components (hierarchical).

The graphical notation and an example can be found on here (<https://www.gliffy.com/examples/uml>)

1e) **Task:** Why should big component-based systems be structured hierarchically?

Solution: From a construction perspective, divide-and-conquer is a well established approach to deal with complex problems. From an analytical perspective, hierarchical decomposition helps to understand abstracted, less complex models and iteratively get more detailed information.

Bibliography

1. Malcolm Douglas McIlroy, *Mass produced software components*. In: Software Engineering: Report of a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7-11 Oct. 1968. Scientific Affairs Division, NATO. Januar 1969. Abgerufen am 10. Oktober 2014..

Task 2: Cheesman/Daniels

The Cheesman/Daniels process helps to identify UML-Components, by stepwise refinement, starting with a requirements specifications.

2a) **Task:** What is a domain model and why is it necessary?

Solution: The domain model (business concept model) describes elements of the (business) domain, their properties and their relationships. It is necessary to capture all entities of the domain, model their relationships to make sure, that the domain and its terminology is understood. It is used as a communication artifact with the stakeholders.

2b) **Task:** What is a business component, according to Cheesman and Daniels?

Solution: A business component consists of business objects (atomic components) and other business components (hierarchical).

2c) **Task:** What should be visible from a component? How is that related to the *Information Hiding Principle*?

Solution: Only implementation independent information should be visible (interfaces). Implementations may change, interfaces are more stable. Furthermore, components with the same interfaces can be exchanged without changing any code.

2d) **Task:** How is the Cheesman/Daniels Process related to technologies like EJB, CORBA, COM+?

Solution: The Cheesman/Daniels Process is a process for developoing (large) component based systems. EJB, CORBA and COM+ are composition systems that can be used as an implementation technology.

Task 3: Designing an Appointment Manager

This task will be used as a basis for the other exercises. You can either solve the task alone or form groups with up to 4 students.

The solution must be send to `christian.piechnick@tu-dresden.de` by June 11th.

You are supposed to develop a component-based appointment management application for the university. At the university, researchers are organized in research groups and work in different projects with other researchers. A researcher has multiple appointments (i.e., research group meetings, research project meetings, visits of guests, talks at conferences, teaching appointments, student meetings). The creator of an appointment can add individual persons, groups or project members. An appointment is shared by all participants, has a title, a location, a start time, a duration and textual notes. Participants can either confirm or decline their presence. Each researcher gets informed (e.g., email, sms) when an appointment was created or changed. Every participant of an appointment can see the status (confirmed, declined) of the other participants.

3a) Task: Design the application following the Cheesman/Daniels process. Create the required models.

3b) Task: Create 1 PDF-File, containing your models in a readable format.

3c) Task: Submit your solution by email to `christian.piechnick@tu-dresden.de`. If you are working in a group, please put the names and matriculation numbers of the individual group members at the first page of your document.