

Examination

Software Engineering (SE)

Prof. Dr.-Ing. Christian Heller <christian.heller@ba-leipzig.de>

Personal Data	
First and Last Name	Ihre Daten müssen von der Klausuraufsichtsperson schriftlich auf der Anwesenheitsliste festgehalten werden!
Matriculation Number	
Subject and Year	CS 2012
Login	Das Login "klaus..." muss unbedingt schriftlich auf der Anwesenheitsliste festgehalten werden, da sonst keine Zuordnung des Logins zu Ihrem Namen und damit keine Korrektur der Klausur möglich ist!

Examination Data	
Date	2014-06-20
Duration [min]	120
Maximum Points [Credit Point]	100
Permitted Study Aids	Dokumentation im lokalen Netzwerkverzeichnis (Intranet); NICHT gestattet sind Kommunikationsmöglichkeiten (Internet) oder Anmeldung via SSH auf dem Rechner "fileserv", wo Ihr Homeverzeichnis liegt, oder eine Anmeldung mit Ihrem Klausur-Login nach Ende der Prüfung. Dies kann leicht geprüft werden (last cs12*, Server-Log-Dateien). Bitte unterlassen Sie also Täuschungsversuche in Ihrem eigenen Interesse.
Remarks	Hinterlegen Sie alle Programme und Antworten in elektronischer Form! Es wird kein Papier angenommen. Möchten Sie Lösungen erläutern, so nutzen Sie Quelltext-Kommentare oder legen eine Text-Datei an. Speichern Sie sämtliche Daten im HOME-Verzeichnis des Nutzers, d.h. unter Windows auf Laufwerk H:\ (NICHT auf C:\ oder "Eigene Dateien")! Idealerweise legen Sie dort ein Unterverzeichnis namens "klausur" an. Lesen Sie die Aufgaben komplett durch, bevor Sie sie lösen! Die Reihenfolge der Lösung ist Ihnen überlassen. Probieren Sie immer, eine Aufgabe zu lösen, da auch auf richtige Teile nicht vollständiger Lösungen Punkte vergeben werden! Falls vom Prinzip her richtig, so werden auch alternative Lösungen akzeptiert. Sie dürfen beliebig viele Bildschirmausgaben von Werten in den Quelltext einbauen, um ein Programm besser nachvollziehen zu können. Diese Aufgabenstellung in Papierform können Sie nach dem Ende der Klausur behalten.

Evaluation											
Task	1	2	3	4	5	6	7	8	9	10	Summe
Desired Value [Credit Point]	20	20	30	10	20	0	0	0	0	0	100

Im Rahmen der Klausur sind fünf Aufgaben verschiedener Themen, immer aber mit Bezug zur *Unified Modeling Language* (UML) zu lösen. Sie sind autonom. Insofern modelliert werden soll, ist *BOUML* als *CASE*-Werkzeug zu verwenden.

Task 1: Use Case Diagram (UCD) [20]

In dieser Aufgabe ist ein Anwendungsfalldiagramm zu erstellen, welches die Abfertigung und Sicherheitskontrolle eines Flughafens erfasst.

a) Verschiedene Arten von *Passagieren* müssen berücksichtigt werden. Neben dem *Reiseleiter* gibt es *Minderjährige* und auch *Behinderte*, mit jeweils speziellen Charakteristika. [4]

b) Alle Passagiere müssen sowohl die *Personenabfertigung*, als auch die *Sicherheitskontrolle* durchlaufen. [2]

c) Der Reiseleiter kümmert sich zusätzlich um die *Gruppenabfertigung*, wobei diese die *Personenabfertigung* voraussetzt. [4]

d) Insofern Gepäck mitgeführt wird, ist die *Personenabfertigung* durch eine *Gepäckaufgabe* zu ergänzen. [2]

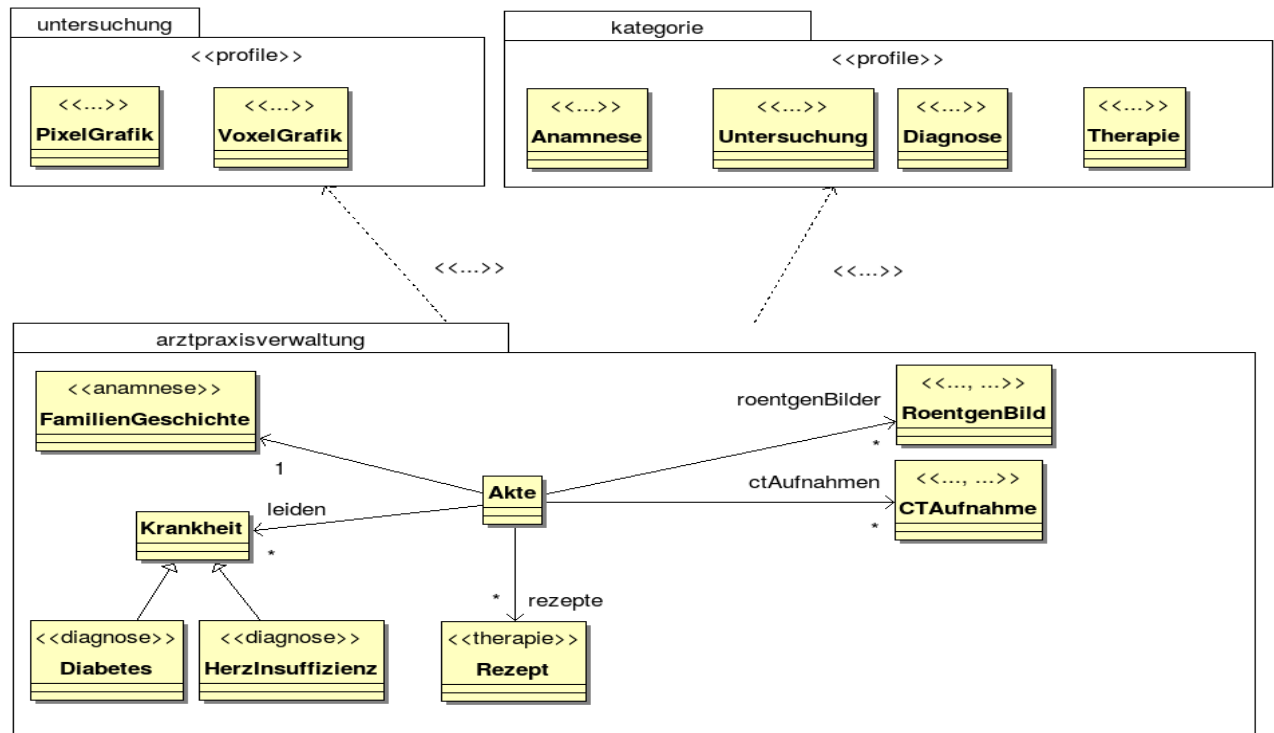
e) Es gibt zwei Möglichkeiten der Personenabfertigung: *Schalter* und *Automat*. [2]

f) Die Sicherheitskontrolle umfasst zum Einen die *Durchleuchtung* des Gepäcks und zum Anderen die Überprüfung mittels *Metalldetektor*. [4]

g) Alle genannten Funktionalitäten gehören zum System *Flughafen*. Die Rollen beteiligter Personen sind als externe Akteure zu verstehen. [2]

Task 2: UML Profile [20]

Gegeben sei folgendes Klassendiagramm:



a) Was versteht man unter einem UML-Profil? [2]

b) Nennen Sie die vier Ebenen der OMG Four Layer Model Hierarchy [4]:

c) In welche der vier Ebenen würden Sie die beiden im Diagramm dargestellten einordnen? [2]

d) Alle Klassen der beiden Profile haben ein-und-denselben Stereotypen. Welchen? [2]

e) Von welcher UML-Metaklasse erben sie? [2]

f) Welcher Stereotyp steht an beiden Abhängigkeitspfeilen? [2]

g) Einem Modellelement können auch mehrere Stereotypen zugeordnet werden. Welche beiden Stereotypen würden Sie den folgenden zwei Klassen zuordnen: *Roentgenbild*, *CTAufnahme*? [2]

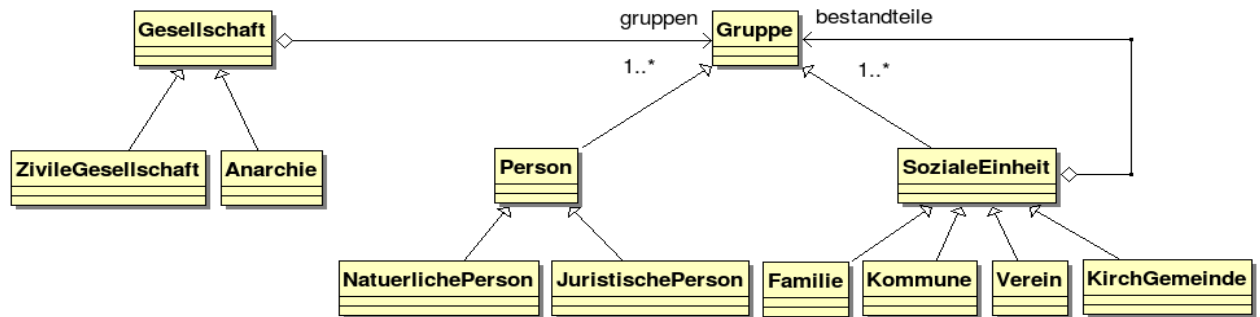
Hinweis: Es passt je ein Stereotyp aus dem Profil "untersuchung" und einer aus "kategorie".

h) Stünde in der Klasse "Voxelgrafik" des Profiles "untersuchung" ein Attribut namens "dimensionen" vom Typ "int", welche Auswirkungen hätte es auf die Anwendung? [2]

i) Wie bezeichnet man solche Vorgaben? [2]

Task 3: Round Trip Engineering (RTE) [30]

Ziel dieser Aufgabe ist die Synchronisation von Modell und Quelltext. Gegeben sei dazu folgendes Klassendiagramm, welches die Struktur von Gesellschaften wiedergibt:



a) Ordnen Sie die gezeigten Klassen wie folgt Quelltext-Artefakten zu [5]:

Artefakt	Klassen
organisationsform	ZivileGesellschaft, Anarchie
ansammlung	Gesellschaft
kategorien	Gruppe, Person, SozialeEinheit
verwaltung	JuristischePerson, Kommune
nichtstaatlich	NaturlichePerson, Familie, Verein, KirchGemeinde

b) Stellen Sie die Artefakte in einem Verteilungsdiagramm dar und machen Sie die durch die Klassen verursachten Abhängigkeiten zwischen ihnen durch entsprechende Pfeile kenntlich! [5]

c) Versehen Sie die Artefakte mit einem Kommentar zu ihrer Beschreibung! Erzeugen Sie eine Dokumentation zum Verteilungsdiagramm im HTML-Format, mit Vektorgrafiken, in ein Unterverzeichnis namens "doc/"! [5]

d) Generieren Sie durch *Forward Engineering* gemäß *Model Driven Architecture* (MDA) den zugehörigen Java-Quelltext mit den üblichen Standardfeldern in ein Unterverzeichnis namens "src/"! [5]

e) Manipulieren Sie den generierten Java-Quelltext, so dass die Klasse *Person* zwei Attribute erhält [5]:

- "name" vom Typ "String"
- "geburtstag" vom Typ "Date"

f) Lesen Sie den geänderten Quelltext ins UML Repository ein, so dass die beiden neuen Attribute der Klasse *Person* im Klassendiagramm sichtbar werden! [5]

Hinweise: Nutzen Sie in BOUML das "Roundtrip-", nicht "Reverse Engineering", letzteres neue Klassen anlegt, statt vorhandene zu ergänzen! Lesen Sie den Quelltext lediglich für die Klasse "Person" ein, nicht für alle Artefakte! Ignorieren Sie einen sich möglicherweise öffnenden Dialog zur Auswahl eines "Java Catalog File", indem Sie <esc> drücken oder <Cancel> klicken! Damit sollte man klar kommen. Wer weitere Hinweise wünscht, findet diese in der Rubrik "Java roundtrip" eines der beiden folgenden Verzeichnisse:

//fileserv/export/info/material/heller_christian/software_engineering/documentation/bouml/doc/

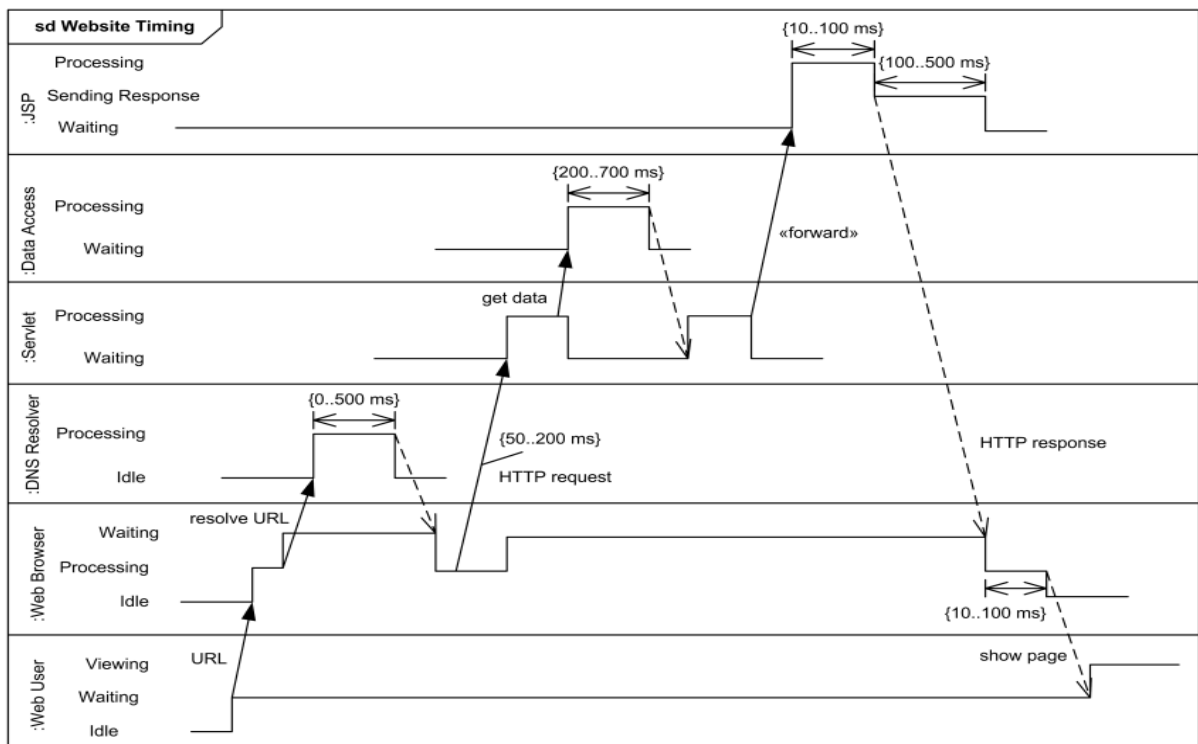
//fileserv/export/info-klausur/heller_christian/software_engineering/documentation/bouml/doc/

Task 4: Timing Diagram (TD) [10]

- Was kann durch das TD ausgedrückt werden? [2]
- Wie heißen die beiden möglichen Darstellungsformen des TD? [2]
- In welche Kategorie in der Systematik der UML-Diagramme würden Sie es einordnen? [2]
- Nennen Sie mindestens zwei Einsatzgebiete! [2]
- In welche Diagramme könnte ein TD transformiert werden? [2]

Task 5: Sequence Diagram (SD) [20]

Gegeben sei ein Zeitverlaufdiagramm (Timing Diagram), welches einige Vorgaben zur Zeitdauer verschiedener Verarbeitungsschritte einer Web-basierten Anwendung macht. Dadurch kann besser abgeschätzt werden, wie lange ein Anwender warten muss, bevor er eine Antwort auf dem Bildschirm sieht.



Erstellen Sie auf der Basis dieses Zeitverlaufdiagrammes ein äquivalentes Sequenzdiagramm! [6 für Objekte; 12 für Nachrichten/Rückgabewerte/Reihenfolge; 2 für Bezeichnung/Rahmen]

Hinweis: Der Fokus sollte auf den gesendeten Nachrichten liegen. Die detaillierten Zeitangaben aus dem Zeitverlaufdiagramm können vernachlässigt werden.

Viel Erfolg!