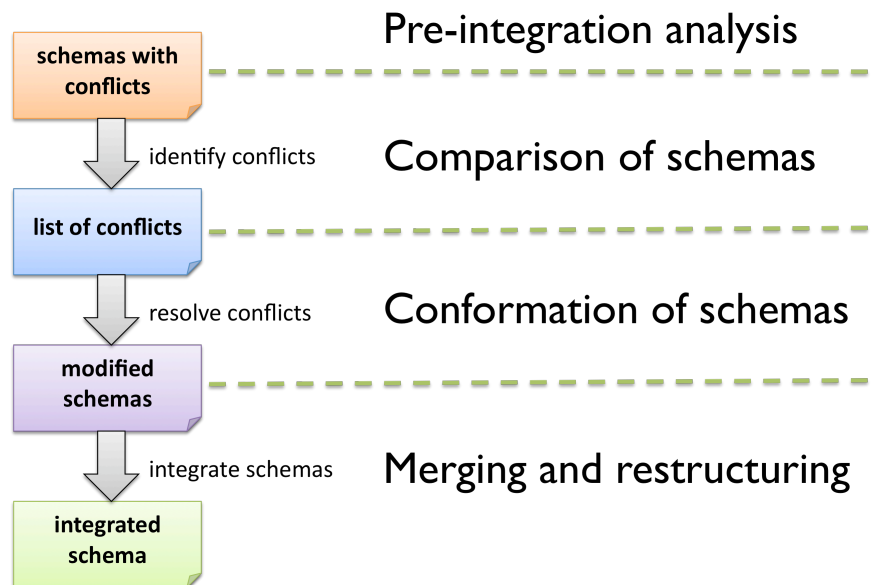


SQL-Lab – Aufgabenblatt 2 – Integration

Allgemeine Informationen

In Aufgabenblatt 1 sollte jede Gruppe entweder [Szenario A: NETFLAX](#) oder [Szenario B: FIRST.FM](#) modellieren. Dieses Aufgabenblatt baut auf den abgegebenen Modellierungen dieser ersten Aufgabe auf. Jede Gruppe bekommt das Diagramm einer anderen Gruppe – die das jeweils andere Szenario modelliert hat – vom Tutor der kleinen Übung zugeteilt und muss es in ihr eigenes Diagramm integrieren. Der Ablauf dieser Integration soll sich formal nach dem in [RDBI Vorlesung 4 \(Folie 15\)](#) vorgestellten Modell richten. Danach erfolgt eine Integration grob in **vier Schritten**:



Der Ablauf, der Inhalt und die Ergebnisse dieser vier Schritte sollten bereits in RDBI im [Aufgabenblatt 4: View Integration \(Aufgabe 4.2\)](#) genauer beschrieben werden. Eine Bearbeitung dieser RDBI Aufgabe ist sehr **hilfreich** für das Verständnis dieses Aufgabenblattes und wird deshalb auch für Teilnehmer empfohlen, die nicht RDBI hören (eine **erneute Abgabe** der RDBI Aufgabe ist natürlich ausdrücklich **nicht nötig**).

Jeder der **vier Schritte** soll ausgeführt und **ausführlich dokumentiert** werden. Dabei ist es besonders wichtig, alle auftretenden Konflikte – etwa Namenskollisionen oder inhaltliche Konflikte – während der Integration zu lösen, damit am Ende ein fehlerfreies und eindeutiges Diagramm entsteht.

Wie bereits bei Aufgabenblatt 1 gilt, dass das fertige UML Diagramm **ausgedruckt** auf einem **DIN A4** Blatt abgegeben werden und **gut lesbar** sein soll. Gefaltete oder nur mit einer Lupe lesbare Abgaben zählen nicht als gut lesbar. Das modellierte Szenario soll von ei-

nem Informatikstudenten im dritten Semester einfach und in der beabsichtigten Art und Weise verstanden werden können. Deshalb gehört zu der Abgabe eine **ausführliche ausformulierte Dokumentation** jedes einzelnen Schrittes der Integration und des resultierenden Diagramms. Die Dokumentation kann z.B. Auszüge des Diagramms enthalten, an denen bestimmte modellierte Aspekte des vollen Diagramms verdeutlicht werden. Um das Diagramm übersichtlicher strukturieren zu können, bietet sich außerdem die Verwendung von **Paketen** an – Containern, die mehrere semantisch zusammengehörige UML Klassen umschließen. Passende **Datentypen** sollen mit modelliert werden, Funktionen nicht. Sinnvolle Ergänzungen und **Kommentare** im Diagramm können eingefügt werden.

Vorbereitung & Werkzeuge

Zur Bearbeitung dieser Aufgabe wird wie schon im ersten Aufgabenblatt ein UML Modellierungstool benötigt. Grundsätzlich darf wieder **jedes geeignete Programm** verwendet werden. Einige mögliche Programme wären:

- ArgoUML (<http://argouml.tigris.org>) – Javabasiert; viele Funktionalitäten
- diagram.ly (<http://diagram.ly>) – Webbasiert; kollaboratives Arbeiten
- Visual Paradigm (<http://visual-paradigm.com>) – Java; Empfehlung eines Teilnehmers

Achtung: In ArgoUML stehen standardmäßig nur wenige Datentypen zur Verfügung (Integer, String, etc). Sollten weitere Typen benötigt werden (wie z.B. Zeit oder Datum) können diese neu definiert werden. Einige in der Vorlesung vorgestellte Features von UML werden von ArgoUML nicht unterstützt (z.B. XOR-Assoziationen oder qualifizierende Assoziationen). Sollte eines der nicht unterstützten Features benötigen werden, kann dies mit einer kleinen Notiz oder frei von Hand direkt im Diagramm annotiert werden.

Liste der Abgaben für Aufgabenblatt 2

Ein **Ausdruck des gut lesbaren UML Diagramms und der ausführlichen ausformulierten Dokumentation auf DIN A4.**