

Dokumentation Task 04

Domain Model and Sequence Diagram

Modul

BTI7081 - Software Engineering and Design

Dozenten

Prof. Dr. Jürgen Vogel

Prof. Künzler Urs

Klasse

I2p

Gruppe

Green

Mitglieder

Andreas Krebs

Cyrill Bolliger

Lorenz Bischof

Martin Scheck

Matthias Keller

Raphael Schmid

Stephan Fahrer

Datum

27. April 2018

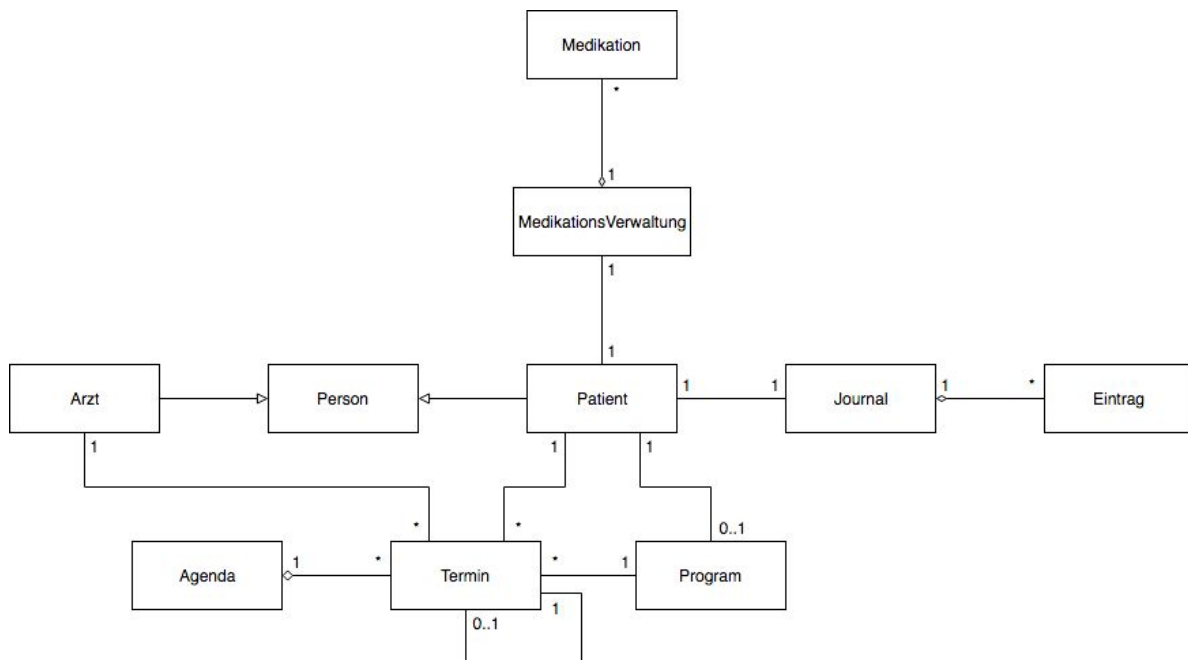
Inhalt

1. Domain Model	2
2. Sequenzdiagramm 1: Eintrag aktualisieren	3
3. Sequenzdiagramm 2: Medikation bearbeiten	4
4. Erweitertes Modell	5

1. Domain Model

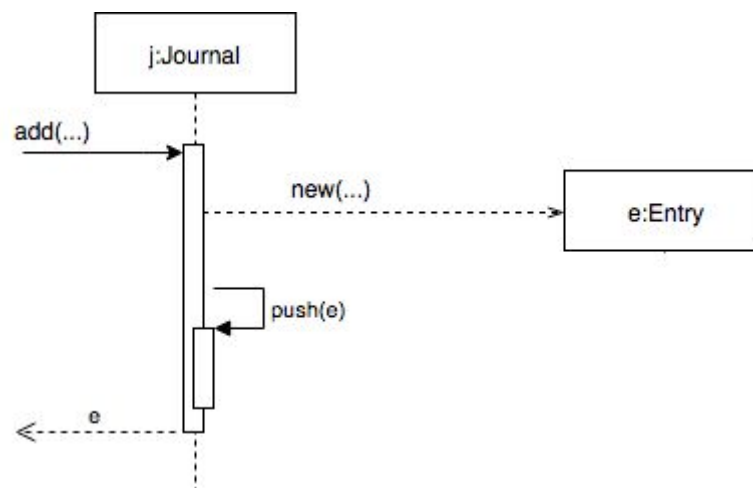
Basierend auf der Responsibility Driven Design Methode, wurde untenstehendes Domain Model entwickelt. Ausgehend von einer natürlichen Person werden Patient und Arzt abgeleitet. Während der Arzt einzig mit Terminen verknüpft ist, wird der Patient zur Ausgangsklasse für Medikation, Journal (Behandlungsakte) sowie Therapieprogramm. Der Patient kann sowohl Termine oder Terminserien haben, die Teil des Therapieprogramms sind als auch besondere Termine (z.B. Sprechstunden), die vom Programm unabhängig sind.

Eine Agenda bildet die zentrale Aggregationsklasse für alle Termine (Singleton) und verantwortet das Terminmanagement (vermeidung von Konflikten).

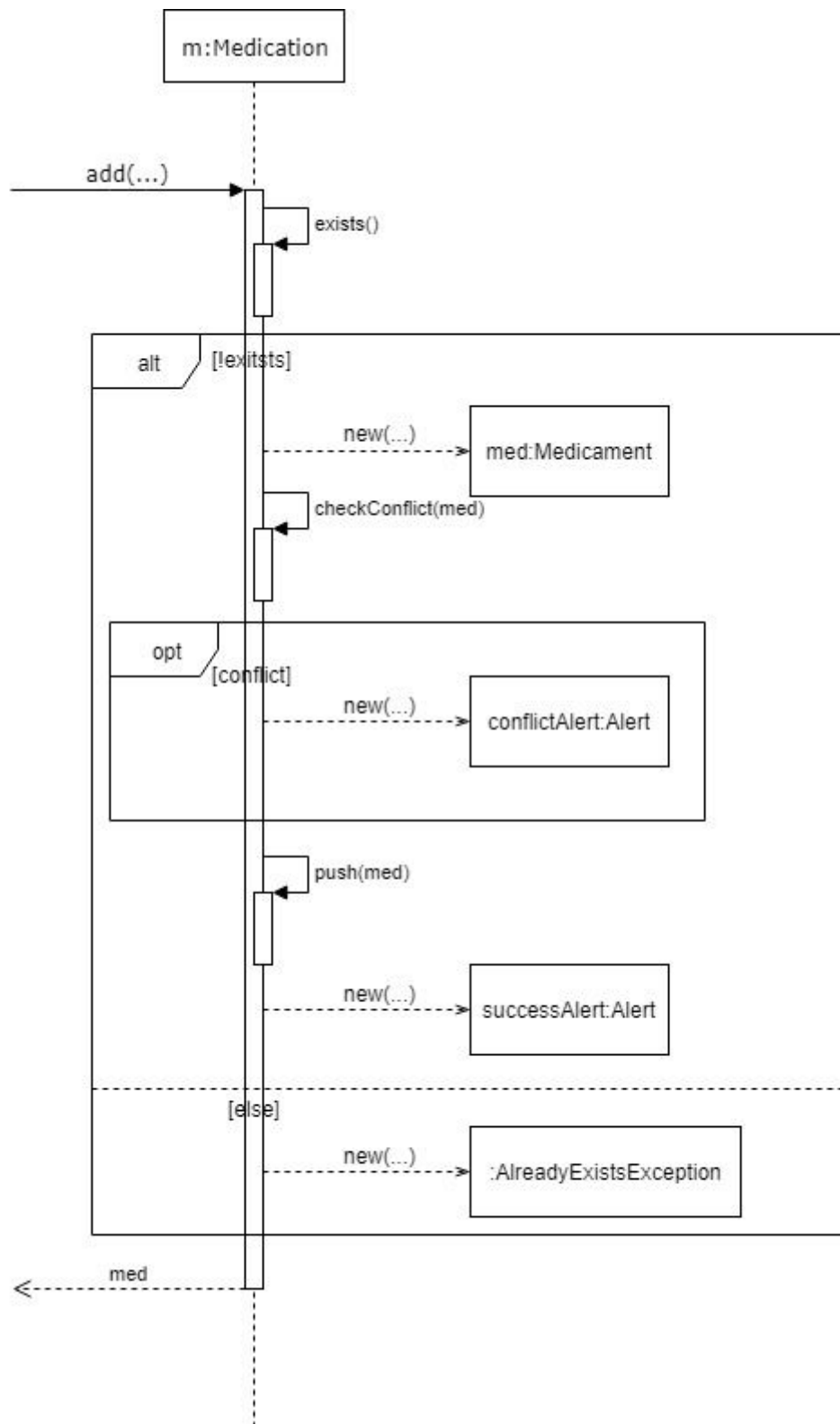


2. Sequenzdiagramm 1: Eintrag aktualisieren

Auf dem Sequenzdiagramm ist der Prozess der Erstellung eines neuen Journaleintrags zu sehen. In der Klasse "Journal" wird durch den Methodenaufruf "add(...)" der Vorgang gestartet, wobei Objekthinhalte wie Textinhalt und Verfasser mitgegeben werden. Das "Eintrag"-Objekt wird erzeugt und anschliessend der Klasse Journal zurückgegeben und eingefügt (Journal ist Liste von "Eintrag"-Objekten).



3. Sequenzdiagramm 2: Medikation bearbeiten



4. Erweitertes Modell

Das folgende Modell zeigt das Domain Model, erweitert um die Klassen, die in obigen Sequenzdiagrammen beschrieben sind.

