Milena Dreier
Dino Buskulic

AUFGABE 2: MINIMALER SPANNBAUM

Entwurf

Komponenten:

Node

Funktionalität Node:

Jede Node eines Graphen ist ein eigener Prozess. Beim Start bekommt jede Node ihren Namen übergeben., mit welchem sie sich anschließend global registriert. Anhand ihres Names lädt die Node ihre Config und speichert die in der Config notierten Edges in ihrer EdgeList. Die EdgeList ist eine Liste aus Tupeln.

Neben der EdgeList setzt die Node zu Beginn noch die Startwerte für ihr Level, BestWeight, FindCount, ihre InBranch, TestEdge, BestEdge und ihren State. Diese Werte werden von dem Algorithmus benötigt und verändert.

Die Schnittstellen die eine Node anbietet sind in der Aufgabenstellung angegeben. Dort ist auch zu vorgegeben, dass bei der Kommunikation zwischen Nodes die Edge. über die die Kommunikation stattfindet, mitgegeben wird.

EdgeList Tupel:

{Weight, NeighbourID, State, NeighbourName}

Weight: Das Gewicht der Edge.

NeighbourID: Die Prozess ID des Nachbarn, der über diese Edge erreicht wird. Über die Prozess ID lässt sich mit dem Nachbarn kommunitzieren.

State: Status der Edge: *basic*, *branch* oder *rejected*. Zu Beginn ist der Status jeder Edge *basic*.

NeighbourName: Der Name des jeweiligen Nachbars. Dient lediglich dem *logging*.

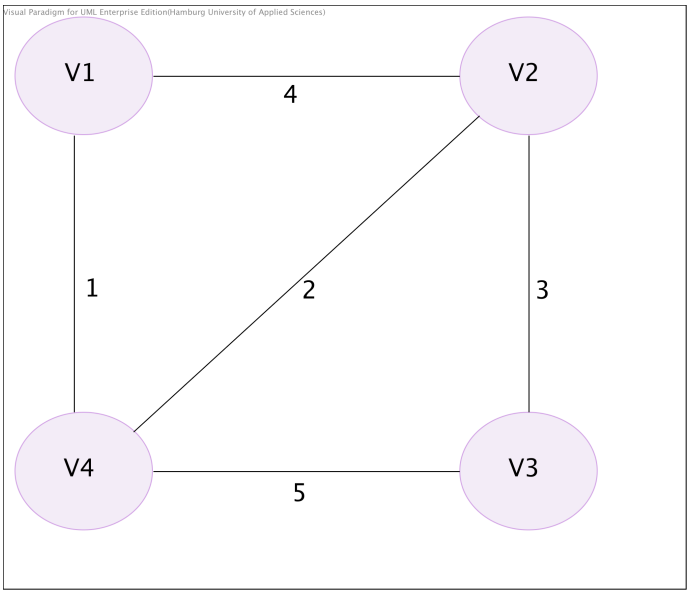
Übergebene Edge:

{Weight, NeighbourFromID, NeighbourToID}

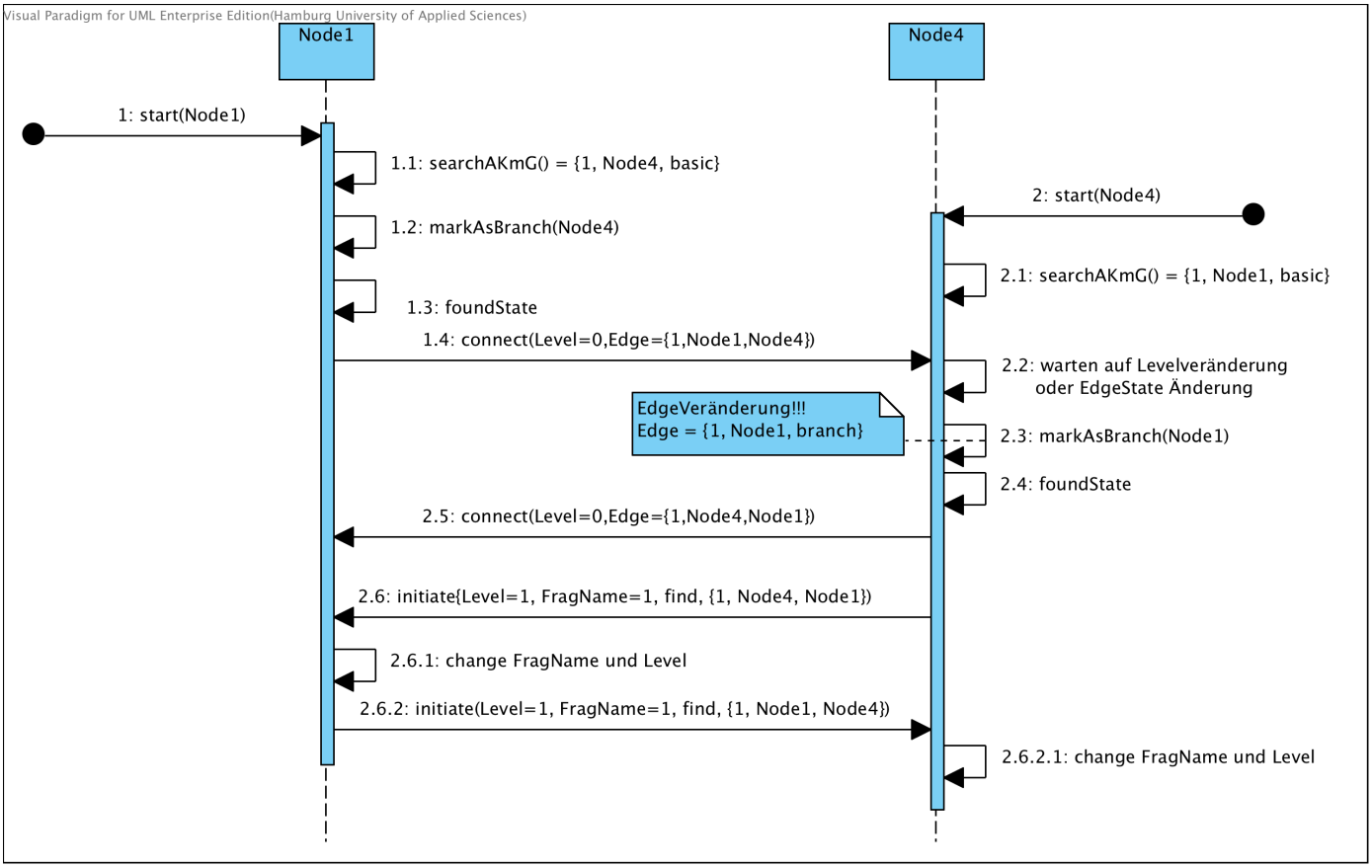
Milena Dreier
Dino Buskulic

Diagramme

Zur Veranschaulichung des Algorithmus haben wir mehrere Sequenzdiagramme angefertigt, die die Kommunikation zwischen den Nodes des folgenden Graphen zeigen.



Sequenzdiagramm bei Start aller Nodes;



Milena Dreier
Dino Buskulic

Diagramm zum Senden von initiate:

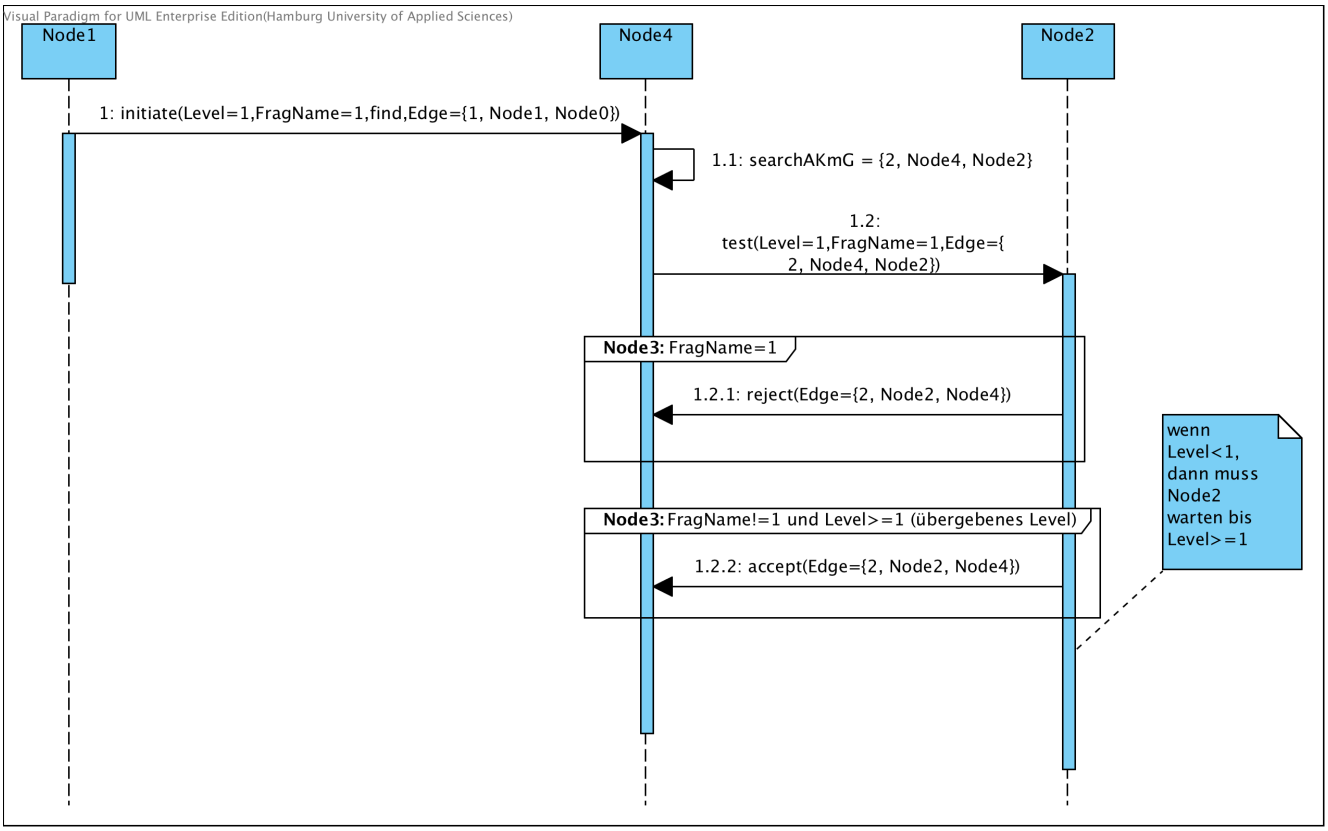


Diagramm zum Senden von report (nicht vollständig, da zu komplex):

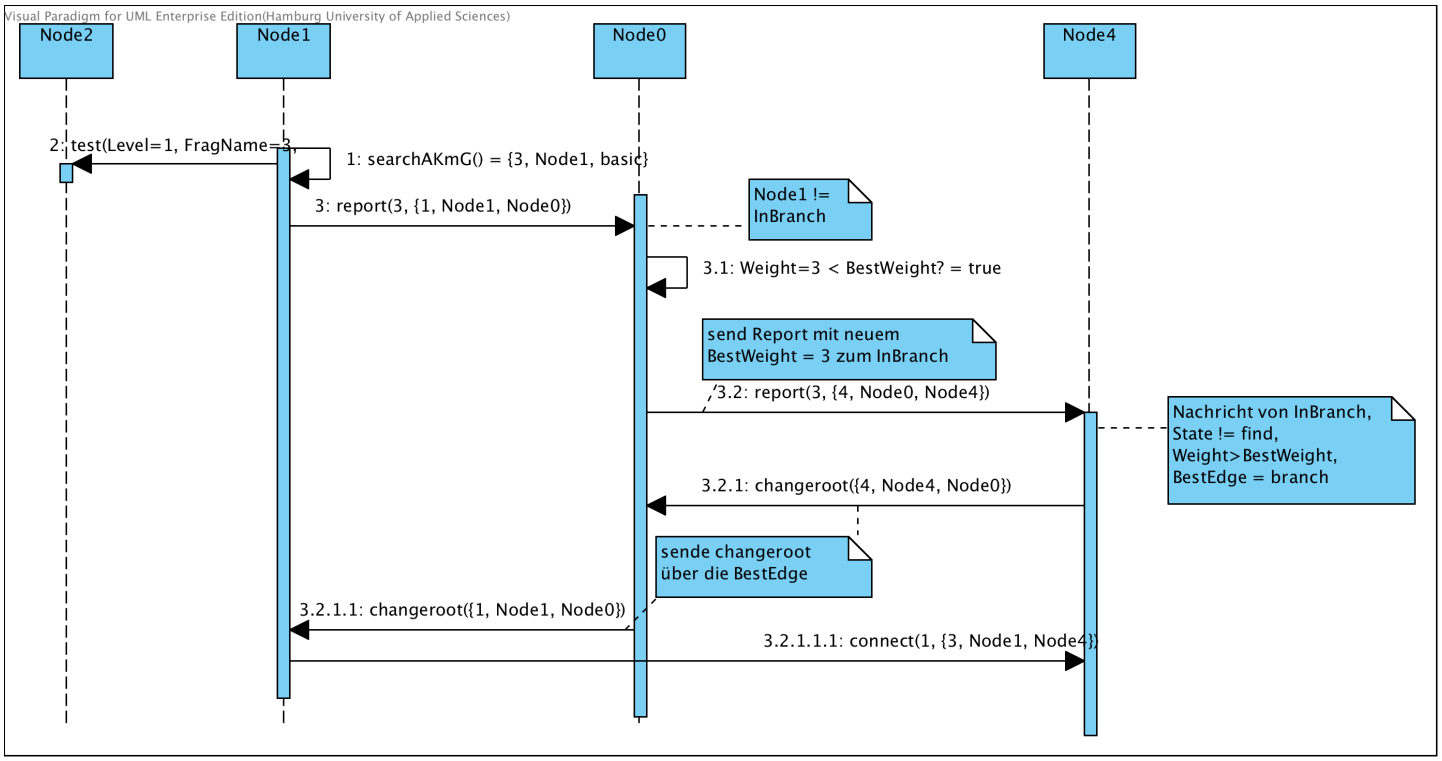


Diagramm mit dem ganzem Algorithmus (nur ein Anfang, da zu komplex):