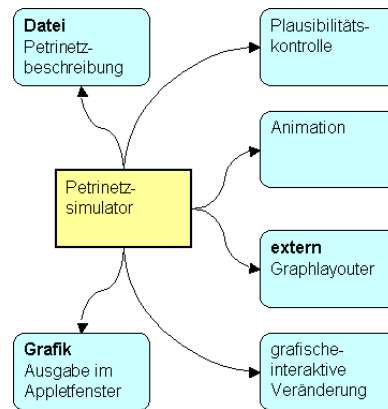


3 Aufgabenstellung

Der Petrinetzsimulator besteht aus insgesamt 7 Modulen, die nach dem Prinzip der hohen Kohäsion entwickelt werden sollen und nach dem Prinzip der schwachen Kopplung miteinander kommunizieren sollen. Die grundlegenden Anforderungen sind:

- Anwendung: Eigenständiges Programm, kein Applet
- Programmiersprache: Java 1.2
- Sprache für grafisches Interface: Java Swing
- Graphlayouter: externes Programm
- Graph-Beschreibungssprache: einheitlich für Laden/Speichern und Graphlayouter

Die Übersicht (rechts) zeigt die einzelnen Module, die unten genauer erläutert werden.



Kontrollmodul „Petrinetzsimulator“ Das Kontrollmodul stellt das Hauptprogramm dar. Es verwaltet die Datenstruktur „Petrinetz“ und steuert die anderen Module des Simulators.

Dateimodul Das Dateimodul ermöglicht das Laden und Speichern des Petrinetzes samt seines derzeitigen Zustandes. Als Beschreibungsformat muss dasselbe Format verwendet werden, das auch für den Graphlayouter erforderlich ist, ggf. mit zusätzlichen Informationen.

Graphlayouter Ein geladenes oder interaktiv erstelltes Petrinetz soll über eine Schnittstelle einem externen Graphlayouter übermittelt werden, der die Positionsangaben für ein (vernünftig) layoutetes Petrinetz berechnet. Verwenden sie wenn möglich „dot“ (siehe Literaturangabe).

Grafikausgabe - Benutzerschnittstelle Der in Abbildung 1 dargestellte Beispielsimulator kann als Orientierung für eine Benutzerschnittstelle dienen. Alle Aktionen zum Erstellen, Layouten, Bearbeiten, Laden, Speichern und Animieren sollen mittels Mausbedienung möglich sein. Es soll u.a. die folgenden Bedienelemente geben:

- Geschwindigkeitsregler zum Einstellen der Animationsgeschwindigkeit (einschl. Einzelschritt)
- Button für: Starten der Animation, Rückwärtslauf der Animation
- Button für: Stop, Herstellen des initialen Zustandes
- Button für: Aufruf des externen Layouters

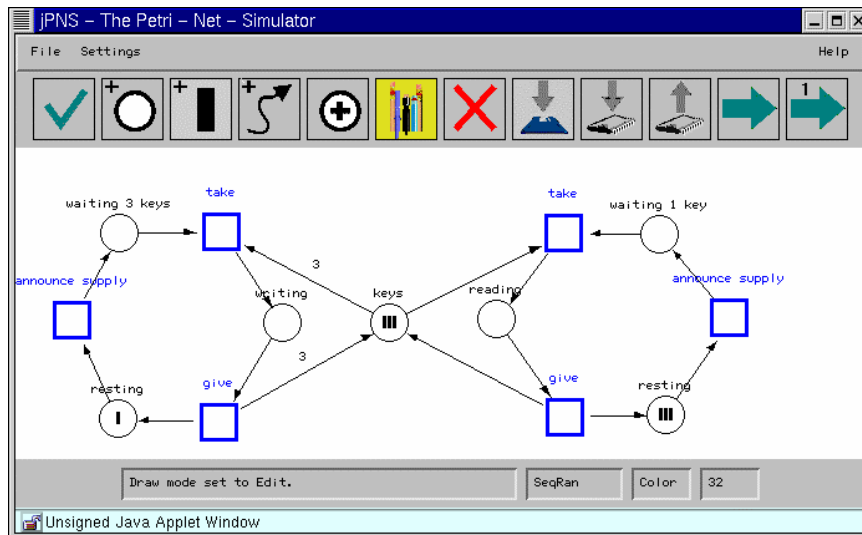


Abbildung 1: Beispiel einer Benutzerschnittstelle

Animation Die Marken sollen sich entsprechend der Petrinetzregeln entlang der Stellen und Transitionen bewegen. Für jeden Animationsschritt soll der Graph nicht vollständig neu gezeichnet werden (Flackereffekt vermeiden).

Interaktion Neben den Aktionen zu den oben erwähnten Bedienelementen sollen die folgenden Aktionen möglich sein:

- Hinzufügen und Löschen von Stellen und Transitionen
- Hinzufügen und Löschen von Kanten
- Hinzufügen und Löschen von Marken
- Gewichtung von Stellen und Kanten
- Beschreibung für Stellen und Transitionen

Plausibilitätskontrolle Prüfen Sie die Gültigkeit des Petrinetzes:

- Verbinden Kanten immer Stellen mit Transitionen (bzw. Transitionen mit Stellen)?
- Liegen keine isolierte Knoten oder Doppelpfeile vor?
- Existiert zwischen zwei Knoten immer nur genau eine Kante?
- Ist keine Stelle für eine Transition zugleich Eingangs- und Ausgangszustand?

Geben Sie während der Animation eine Warnung aus, wenn Konflikte auftreten. Ein Konflikt liegt immer dann vor, wenn unklar ist, welche Transition schalten darf bzw. wenn gar keine Transition mehr schalten kann.

4 Organisatorisches

Woche	Datum	Mo	Di	Mi	Do	Fr	Abgabe (A), Kolloquium (K)
1	01.05						
2	08.05				A		Pflichtenheft (wenn möglich mit \LaTeX)
3	15.05		K				
4	22.05						
5	29.05				A		Entwurf & Spezifikation
6	05.06		K				
7	12.06	Pfingstwoche					
8	19.06						
9	26.06						
10	03.07				A		Implementierung
11	10.07		K				
12	17.07				A		Testbericht & Demo
13	24.07		K				
14	31.07						

Abbildung 2: Termine (vorläufig)

Die Zeiteinteilung für das Softwareengineering-Praktikum ist in Abbildung 2 zu sehen. Zu jeder Phase muss das Phasendokument in zwei Exemplaren zum entsprechenden *Donnerstag* bis 14:00 Uhr beim Betreuer abgegeben werden. Der rechte Rand sollte 5cm breit sein. Am darauffolgenden Dienstag findet mit jeder Gruppe ein Kolloquium statt, in dem der jeweilige Phasenverantwortliche die Phase mit einem Kurzvortrag (ca. 15 Minuten) vorstellt. Die Kolloquien beginnen um 10:00; die genauen Zeiten werden mit den einzelnen Gruppen festgelegt.

5 Literatur

1. Prof. Snelting: Vorlesungsskript „Software-Engineering“
2. Prof. Balzert: Lehrbuch der Software-Technik, Bd. 1 & 2, Spektrum
3. Prof. Balzert: Lehrbuch der Objektmodellierung, Spektrum
4. Oestereich: Objektorientierte Softwareentwicklung, Spektrum
5. Dr. Reisig: Systementwurf mit Netzen, Springer
6. Dr. Reisig: A Primer in Petri Net Design, Springer
7. Graphlayouter „dot“: <http://www.research.att.com/sw/tools/graphviz/>
8. Swing: <http://java.sun.com/docs/books/tutorial/uiswing/index.html>
9. Swing: <http://www.mut.de/lesecke/buecher/java2/inhalt.html>
10. Walrath, Campione, The JFC Swing Tutorial, SUN
11. Eckstein, Loy, Wood: Java Swing, O'Reilly
12. Drye, Wake, Java Foundation Classes - Swing Reference, Manning