

Department of Computer Sciences
University of Salzburg

Einführung Simulation
SS 13/14

Notaufnahme

July 28, 2014

Project Members:

Tobias Auinger, 1220321, auingerto@stud.sbg.ac.at
Christian Müller, 1123410, mueller110@gmx.net
Georgi Potzkov, 0123456, potzkovge@stud.sbg.ac.at

Academic Supervisor:

asdf
asdf

Contents

1	Einleitung	2
2	Aufgabe	2
3	Erweiterungen	2
4	Events und Warteschlangen	2
4.1	New Patient Event	2
4.2	Patient Arrival Event	3
4.3	Treatment Termination Event	3
4.4	Busy/Free Doctor Warteschlangen	4
5	Ergebnisse	4

1 Einleitung

In folgender Ausarbeitung sollen Modell und Ergebnisse der von uns im Rahmen des Proseminars Einführung Simulation implementierten Projektes besprochen werden. Ziel des Projektes war es eine auf DesmoJ basierende Simulation einer Notaufnahme zu erstellen. Wir haben uns für den Ansatz der Ereignisorientierten Simulation entschieden dh. Zustandsänderungen erfolgen stets nur zu bestimmten Ereigniszeitpunkten.

2 Aufgabe

Unsere Aufgabe war es eine Notaufnahme eines Krankenhauses zu simulieren. Die Notaufnahme wird von 2 Ärzten betreut. In dieser Notaufnahme erscheint durchschnittlich alle 40 Minuten ein Patient. Der ankommende Patient hat eine Priorität, die seine Dringlichkeit beschreibt. Etwa 20% der ankommenden Patienten haben die Priorität 3 und müssen so schnell wie möglich behandelt werden. Sie sind also akute Notfälle. Der Rest, also Patienten mit der Priorität 1, kann warten. Nach einer Behandlung eines Patienten, wird dessen Priorität auf 2 gesetzt. Diese Patienten werden noch einmal für eine Nachbehandlung gebraucht, bevor sie die Notaufnahme endgültig verlassen. Zu simulieren war ein längerer Zeitraum, wie etwa 20 Tage. Die Notaufnahme hat dabei immer offen zu sein. Ebenso war es unsere Aufgabe Erweiterungen zu überlegen und zu implementieren.

3 Erweiterungen

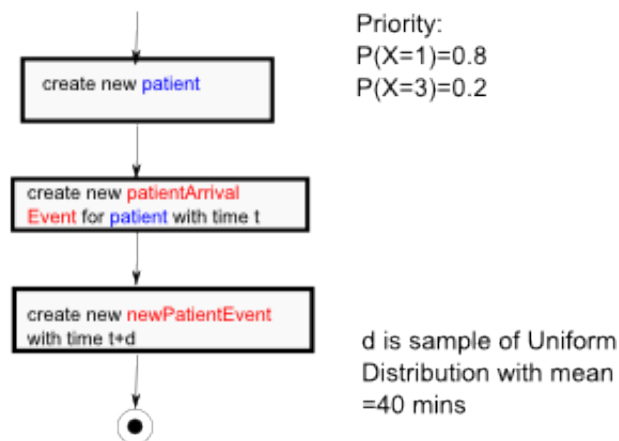
Wie schon in 2 angesprochen, sollten wir ebenfalls ein paar Erweiterungen überlegen und implementieren. Um ein bisschen Realitätsnäher zu sein, dachten wir uns, dass dringendere Notfälle (Patienten mit Priorität 3) nicht so dringende Patienten (Patienten mit Priorität 1) verdrängen können, da diese eventuell schwerwiegende, also lebensbedrohende, Verletzungen haben können. Desweiteren dachten wir uns, dass akute Notfälle auch versterben können. Somit sind wir auf den Entschluss gekommen, den Patienten mit der Priorität 3 eine zufällige Restzeit zu geben, in der sie behandelt werden müssen. Werden sie jedoch nicht behandelt, versterben die Patienten und werden dadurch auch aus der Notaufnahme gebracht.

4 Events und Warteschlangen

4.1 New Patient Event

Um einen neuen Patienten zu erzeugen verwenden wir unser New Patient Event. Nun wird eine neue Patient Entität erstellt. Hierbei bekommt der Patient mit einer Wahrscheinlichkeit von 20

NewPatientEvent



4.2 Patient Arrival Event

PatientArrivalEvent modelliert die Ankunft eines Patienten. Abhängig von der Priorität des Patienten wird dieser in einer der 3 Warteschlangen gegeben. Wurde der Patient im Sinne unserer Erweiterung von einem Patienten mit Priorität 3 abgelöst dh. wurde seine Behandlung durch einen Priorität 3 abgebrochen, wird dieser in vorne in die Warteschlange seiner Priorität eingereiht. Handelt es sich bei dem Patienten um einen Patienten der Priorität 3 wird für diesen weiter die Zeitpunkt seines Todes bestimmt. (PatientDeathEvent) Abhängig davon ob ein Doktor frei ist dh. gerade keinen Patienten behandelt oder nicht wird der Patient einem Doktor zugewiesen oder aber er verbleibt in der Warteschlange. Handelt es sich beim Patienten um einen Notfall hat dieser auch wenn kein Doktor frei sein sollte durch Ablösung eines sich in Behandlung befindlichen Priorität 1 Patienten die Möglichkeit sofort behandelt zu werden.

4.3 Treatment Termination Event

Das Treatment Termination Event ist sowhol ausschlaggebend für den weiteren Verlauf in der Notaufnahme des Patienten, als auch für die Dauer seiner Behandlung. Der Patient wird aus der Behandlungswarteschlange genommen. Ist dessen Priorität nicht 2 (sondern 3 oder 1), dann wird seine Priorität auf 2 gesetzt und anschließend ein Patient Arrival Event für diesen erstellt. Ist jedoch seine Priorität 2, so verlässt er die Notaufnahme. Beinhalten nun alle Warteschlangen der Patienten keine Patienten, so wird der Doktor (der diesen Patienten behandelte) auf verfügbar gesetzt, in dem er der Free Doctor Warteschlange hinzugefügt wird. Enthält jedoch eine der Patienten Warteschlangen einen Patienten (Überprüfung erfolgt absteigend (3,2,1)), so wird dieser aus der Warteschlange entfernt und wird von einem Doktor behandelt (Hinzufügen der Treatement Warteschlange). Ebenso wird ein neues Treatement Termination Event für den neuen Patienten mit einer Behandlungszeit erstellt und der Ereignisliste hinzugefügt. Hat dieser neue Patient eine Priorität von 3, so wird sein Patient Death Event, welches den Tod für diesen nicht behandelten akuten Notfall darstellt, abgebrochen.

4.4 Busy/Free Doctor Warteschlangen

Die Doktoren in unserer Notaufnahme werden mittels zwei Warteschlangen verwaltet. Diese Graphik beschreibt den Statuswechsel der Doktoren.



5 Ergebnisse

Im folgenden sollen die Ergebnisse, welche das Resultat von jeweils hundert Simulationsläufen sind, einer näheren Betrachtung unterzogen werden. Wie in 2 beschrieben, war es Teil der Aufgabenstellung die Simulation mit und ohne Initialisierungsphase durchzuführen. Desweiteren haben wir die Auswirkung der von uns erdachten Erweiterungen betrachtet und analysiert.