9. Symposium über Simulation als betriebliche Entscheidungshilfe
15. – 17. März 2004 in Braunlage (Harz)

Ein Verfahren zur Optimierung von Geschäftsprozessen mit attributierten Petri-Netzen

Bernd Eichenauer, IBE Simulation Engineering GmbH

Abstract: Es wird ein Verfahren vorgestellt, mit dem sich Geschäftsprozesse unter weitgehend beliebig vorgebbaren Einschränkungen exakt modellieren und optimieren lassen.

In letzter Zeit findet die Optimierung von Geschäftsprozessen in der industriellen Praxis zunehmend Beachtung, weil hier mit vergleichsweise geringen Mitteln große Kosteneinsparungen erzielt werden können. Außerdem handelt es sich um eine der letzten noch verfügbaren organisatorischen Möglichkeiten, um dem durch die Globalisierung der Märkte hervorgerufenen Kostendruck zu begegnen. Im folgenden wird eine Modellierungsmethode vorgestellt, mit der sich exakte, ausführbare Modelle (Simulationsmodelle) für diskrete Prozesse erstellen lassen, die auch den Einsatz mathematischer Optimierungsverfahren ermöglichen. Die Methode hat sich bei der Modellierung zahlreicher Anwendungen bewährt (siehe z.B. [1,2,3,4,5,6,7,8]).

1. Die Modellierungsmethode

Planungssysteme für Workflow- und Automatisierungsvorhaben basieren sehr häufig auf den CASE-Methoden der 80-iger und 90-iger Jahre und sind bis heute bei der Systementwicklung nur beschränkt hilfreich, weil sie die zu planenden Systeme meist nur unvollständig beschreiben und verifizieren. Häufig werden nur die statischen Teile einer Anwendung unter Verwendung von halbformalen Systemspezifikationen mit Pseudocode, Programmcodeeinschüben und mit zahlreichen graphischen Darstellungen beschrieben. Geprüft wird dabei höchstens die Konsistenz der Spezifikation, nicht deren Inhalt. Gerade die Aufgaben, bei denen der Computer hilfreich sein könnte, nämlich die Verifikation der dynamischen Angaben in einer Systemspezifikation, werden durch heute verfügbare CASE-Tools unzureichend unterstützt.

Ein weiterer Nachteil dieser Beschreibungsmethode ist die Diskrepanz zwischen der visuellen Sicht auf die Anwendung und der Abstrahierung im Modell, die besonders bei Einsatz objektorientierter Methoden häufig zu schwer verständlichen Modellen führt. Die Folge ist mangelnde Akzeptanz und fehlende Verwendung durch die Anwender, für welche die Modelle eigentlich erstellt wurden.

Diese und einige weitere Mängel waren der Anstoß für die Suche nach einer anwenderfreundlichen und gleichzeitig exakten Modellierungsmethode. Ausgangspunkt war dabei die Forderung, dass ein Anwender seine Anwendung im Modell wiedererkennen soll, das Modell realitätsnah, d.h. sowohl strukturell als auch funktionell ein Abbild dieser Anwendung sein soll.

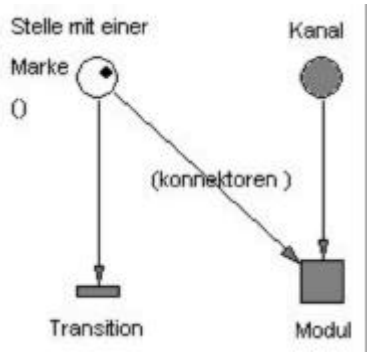
Strukturelle Übereinstimmung zwischen Realität und Modell lässt sich erreichen, wenn die Modellierungsmethode es ermöglicht, die visuelle Struktur einer Anwendung eins zu eins in ein Modell zu übertragen. Funktionell identisches Verhalten impliziert, dass das Modell aus-

führbar, d.h. ein Simulationsmodell sein muss. Ein ausführbares (animierbares) Modell ermöglicht kostengünstige Experimente, mit denen zum Beispiel der Systementwurf verifiziert oder die optimale Parametrierung hinsichtlich bestimmter Kriterien (Produktionskosten, usw.) gefunden werden kann.

Die Analyse der für die exakte Modellierung von industriellen Anwendungen erforderlichen Beschreibungsmittel führte zur Entwicklung einer halbgrafischen Modellierungssprache MSL[1] und zu deren Implementierung in der Entwicklungsumgebung PACE [9]. MSL bietet drei Arten von Sprachelementen:

- Sprachelemente zur Beschreibung der Struktur einer Anwendung samt der Prozessbahnen in dieser Anwendung.
- Sprachelemente zur Darstellung der Objekte, die auf den Prozessbahnen bewegt werden.
- Sprachelemente für die Verarbeitung der Objekt- und sonstigen Prozessdaten.

Sprachelemente zur Darstellung von Prozessbahnen kann man von den Petri-Netzen übernehmen. In MSL werden die Verarbeitungsschritte mit verallgemeinerten Stellen, Transitionen und Konnektoren beschrieben. Die Verallgemeinerung besteht darin, dass die Petri-Netz-Elemente mit zahlreichen Attributen versehen werden, welche die genaue Verarbeitung der Objekte an den Netzknoten festlegen.



Außer den aus S/T-Netzen bekannten Netzelementen enthält MSL zwei weitere Netzelemente für die hierarchische Strukturierung von Netzen, die als "Modul" und "Kanal" bezeichnet werden. Mit ihnen kann die Systemstruktur des realen Systems im Modell abgebildet werden. Module enthalten wiederverwendbare Teilnetze, die über Stellen und Kanäle als Schnittstellen in ein beliebiges Netz eingefügt werden können. Kanäle sind wie Stellen passive Netzelemente und werden als Zusammenfassung verwendet, wenn Module über mehrere Konnektoren verbunden sind.

Die zu bearbeitenden Objekte werden im Modell, ebenfalls in Anlehnung an Petri-Netze, durch Marken oder Behälter dargestellt, die auf den Prozessbahnen durch das Netz laufen und die im Modell benötigten charakteristischen Eigenschaften der realen Objekte als Attribute tragen. Damit die Objekte in den Transitionen ansprechbar sind, werden ihren Attributen durch Attributierung der Konnektoren sog. Konnektorvariablennamen zugeordnet. Die Bearbeitung (Veränderung) der Objekte und der sonstigen Prozessdaten wird hauptsächlich in den Transitionen vorgenommen, die Anweisungsfolgen in einer geeigneten Script- oder Programmiersprache enthalten. Derzeit wird in MSL für die Attributierung Smalltalk-80 verwendet.

Das folgende einfache Beispiel eines Autowaschplatzes zeigt das Zusammenspiel der verschiedenen Sprachelemente. Man erkennt die von S/T-Netzen her bekannten Stellen und Transitionen, die teilweise mit Attributen versehen sind, welche die Zuführung von Fahrzeugen ins Modell festlegen, die Dauer des Waschvorgangs regeln oder die Anzahl der gewaschenen Fahrzeuge zählen und ausgeben. Das "Distribution Time Histogram" zeigt das Ergebnis des Simulationslaufs, d.h. die Wahrscheinlichkeiten für das Auftreten von 0, 1, 2,... Fahrzeugen in der Fahrzeugwarteschlange "Warten" an.

[1] Modeling and Simulation Language

2. Netzfunktionen

Die Attributierung von Netzelementen ermöglicht zahlreiche neuartige Netz-Konstruktionen. Dabei sind abgeschlossene Prozessbahnen von besonderer Bedeutung, die von verschiedenen Transitionen her mit unterschiedlichen Parametern "aufgerufen" werden können und die Ergebnisse an die aufrufende Umgebung zurückgeben. Solche Prozessbahnen werden im folgenden als "Netzfunktionen" bezeichnet. Zunächst wird eine Konstruktion betrachtet, die für die Erstellung solcher Funktionen verwendet werden kann.

In realen Systemen treten neben synchronen Ereignissen auch asynchrone Ereignisse auf. Letztere sind in konventionellen Petri-Netzen nicht vorgesehen. In MSL werden asynchrone Ereignisse durch Einlegen einer Marke, wahlweise mit Attributen, in eine Stelle erzeugt. Dieser Vorgang wird in MSL mit einer addTokenTo:-Botschaft programmiert.

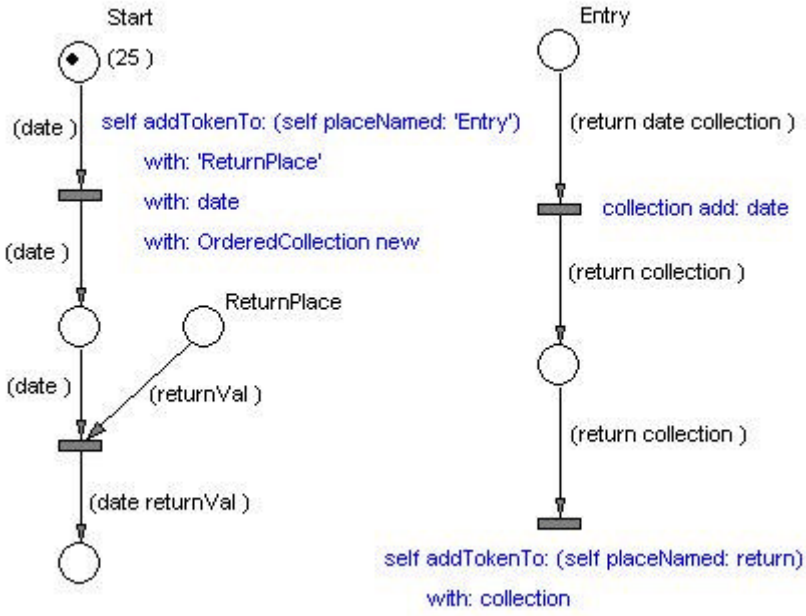
Beispiel:

Die Abbildung zeigt zwei einfache Prozessbahnen mit einer Initialmarke auf der linken Bahn.

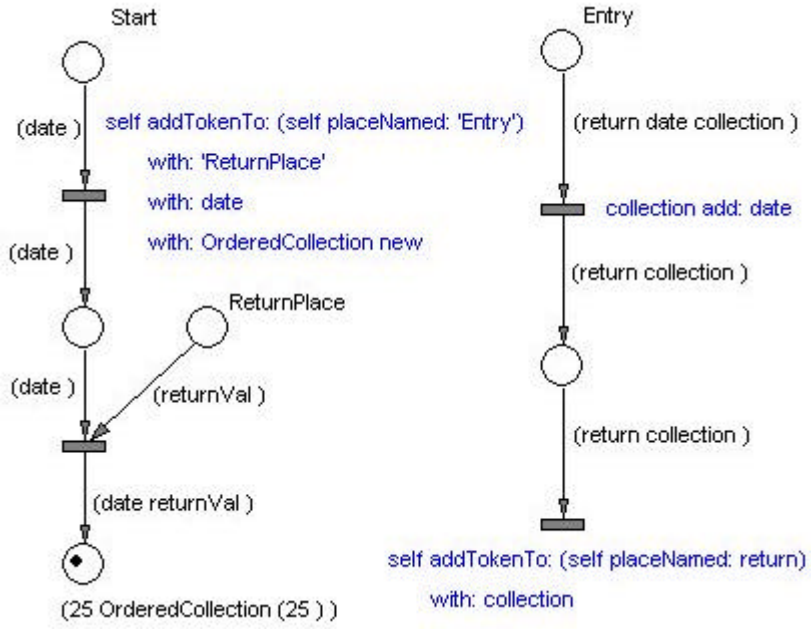
Bei der Ausführung des Netzes wird zunächst ein Prozess auf der linken Bahn gestartet, der eine Marke mit Parametern in die Stelle "AsynEreig" legt und dadurch einen Prozess auf der rechten Bahn startet. Der Endzustand ist in der folgenden Abbildung dargestellt.

Der beschriebene Mechanismus kann nun wie folgt für die Konstruktion von Netzfunktionen verwendet werden: als Parameter für die Netzfunktion wird neben gegebenenfalls zu übergebenden Anwenderdaten auch der Name der Stelle übergeben, in der das Ergebnis abgeliefert werden soll. Für die Rückgabe des Funktionswerts wird wieder eine addTokenTo:-Botschaft verwendet.

Beispiel:



Die rechte Prozessbahn stellt eine Netzfunktion dar, die von der linken Bahn mit drei Parametern beauftragt wird und ein Ergebnis in die Stelle "ReturnPlace" legt. Das Ergebnis zeigt das folgende Netz:



3. Optimierer

Mit der Verfügbarkeit von Netzfunktionen entsteht sofort die Frage nach den Extremwerten dieser Funktionen. Denken wir uns nämlich zu optimierende Teile von Geschäftsprozessen in Form von Netzfunktionen dargestellt, so geben die Extremwerte dieser Netzfunktionen gerade die gewünschte optimale Parametrierung bzw. Auslegung dieser Teilprozesse an.

Die Bestimmung der Extremwerte der Netzfunktionen kann in voller Analogie zur Bestimmung der Extremwerte mathematischer Funktionen erfolgen. Bei der Auswahl der Optimierungsverfahren ist allerdings zu beachten, dass die analytischen Eigenschaften der Netzfunktionen im allgemeinen nicht bekannt sind. In vielen Fällen handelt es sich um verallgemeinerte Funktionen, die nur für bestimmte Argumentwerte berechenbar sind. Dieser Fall tritt häufig bei der Optimierung von nicht teilbaren Ressourcen auf.

Bei der Auswahl der Optimierungsverfahren wurden deshalb nur Verfahren ausgewählt, bei denen die Stetigkeit der Netzfunktionen ausreicht. Für die Fälle, in denen das nicht der Fall ist, wurden die Verfahren durch die zusätzliche Angabe einer Skalierung erweitert. In der MSL-Entwicklungsumgebung PACE sind derzeit das Aufstiegsverfahren, das Simplex-Verfahren und ein genetisches Verfahren für mathematische Funktionen und für Netzfunktionen implementiert. Die Verfahren können einzeln und in Sequenz verwendet werden.

Um die Optimierer anzuschließen, ist der beschriebene Aufrufmechanismus für Netzfunktionen geringfügig zu ändern. Der Funktionsaufruf erfolgt nicht mehr über eine addTokenTo:-Botschaft, sondern über den Aufruf des Optimierers, bei dem die Eingangsstelle der Netzfunktion und die Ergebnisstelle für den Extremwert anzugeben ist. Für die Rückkehr aus der Netzfunktion wird die netResult:-Botschaft verwendet, welche den aktuellen Funktionswert an den aufrufenden Optimierer liefert.

Beispiel:

Das Beispiel berechnet, ausgehend vom Anfangswert 0.2 nach dem Aufstiegsverfahren das Maximum der Sinusfunktion.

4. Fallstudie

4.1 Aufgabenstellung

Das beschriebene Optimierungsverfahren soll im folgenden an einem etwas umfangreicheren Beispiel demonstriert werden. Betrachtet wird eine Produktionsanlage, die in zwei Abschnitte, nämlich in eine Arbeitsvorbereitung und eine Fertigung unterteilt ist. Hergestellt werden fünf Produkte, deren Produktionsparameter in der folgenden Tabelle aufgelistet sind:

ProduktNr	Vorbereit	Fertigung	Mittelwert	Spanne
1	2	2.8	5	400
2	1.5	2.4	3	420
3	1.2	2.5	7	380
4	1.8	2.1	1	500
5	2.5	4.5	4	560

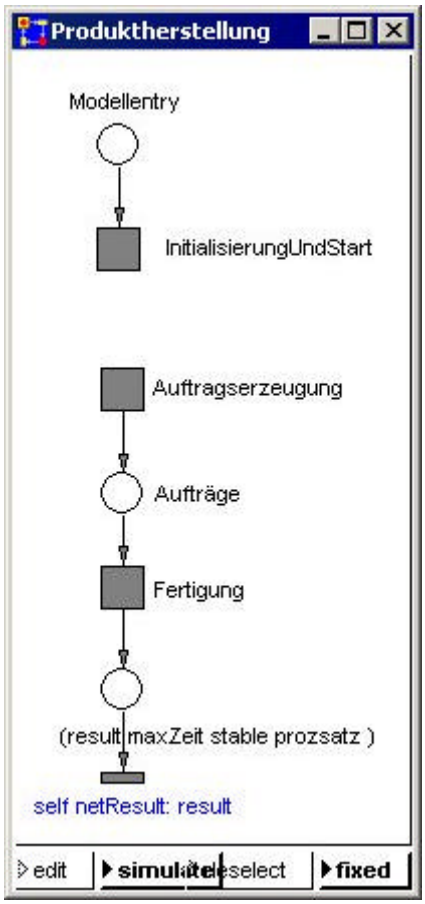
Die Spalten geben der Reihe nach die während der Simulation auszuwertenden Parameter an:

Spalte 1: Produktnummer
Spalte 2: Zeit in Stunden, die für die Vorbereitung der Fertigung benötigt wird.
Spalte 3: Zeit in Stunden, die für die Fertigung benötigt wird.
Spalte 4: Mittlere Zeit in Stunden zwischen zwei Verkäufen.
Spalte 5: Verdienstspanne = Verkaufspreis - Materialkosten
ohne Bearbeitungskosten (z.B. in €).

Weiter seien die Arbeitskosten pro Stunde für einen Arbeitsvorbereiter 30 €, für einen Fertiger 40 €.

4.2 Modellierung

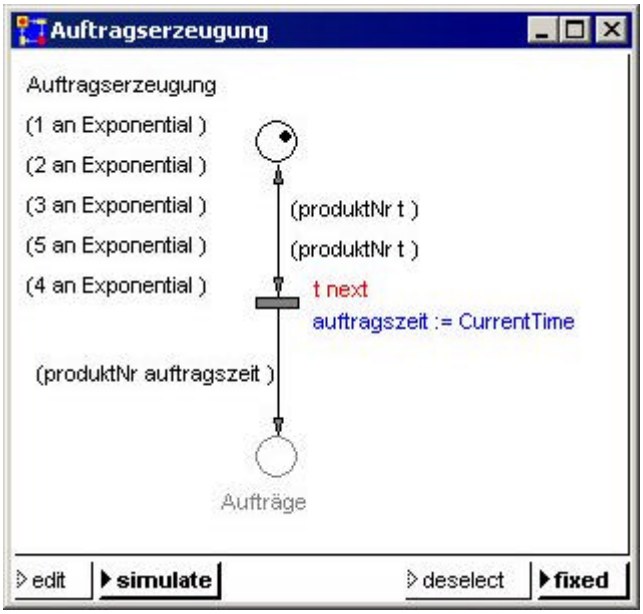
Die nebenstehende Abbildung zeigt die oberste Hierarchieebene der Anwendung als Netzfunktion. Tiefer liegende Ebenen, die sich durch Öffnen der Module anzeigen lassen, enthalten zunehmend Implementationsdetails und werden normalerweise bei der Erstellung von Anwendungssimulatoren nicht gezeigt.



Die Funktion besteht aus zwei Teilen, dem Modul "InitialiserungUndStart" und dem Teil, in dem die Produktion modelliert ist. Dieser besteht aus der Abteilung "Auftragserzeugung" und der "Fertigung". Die Funktion wird durch Einlegen einer Marke in die Stelle "Modellentry" mit zwei Parametern beauftragt. Übergeben wird die Anzahl der Facharbeiter in der Arbeitsvorbereitung und in der Fertigung. Funktionsergebnis ist der innerhalb einer vorgegebenen, hinreichend großen Zeit (z.B. 1000 Stunden) anfallende Gewinn.

Die Initialisierung braucht hier nicht detailliert betrachtet zu werden. Sie sorgt dafür, daß beim Start der Produktion immer die gleichen Anfangsbedingungen vorliegen und legt die Besetzung den beiden Abteilungen mit Facharbeitern aufgrund der Aufrufparameter fest. Interessant ist der Start der Auftragserzeugung, der durch Einlegen von 5 Marken in die Stelle "Auftragserzeugung" des gleichnamigen Moduls bewirkt wird. Entsprechend wird die Erzeugung von Aufträgen durch Löschen der Marken in der Stelle "Auftragserzeugung" beendet. Auf diese Weise lassen sich mit der addTokenTo-Botschaft und der Botschaft zum Löschen aller Marken in einer Stelle relativ einfach Schalter modellieren.

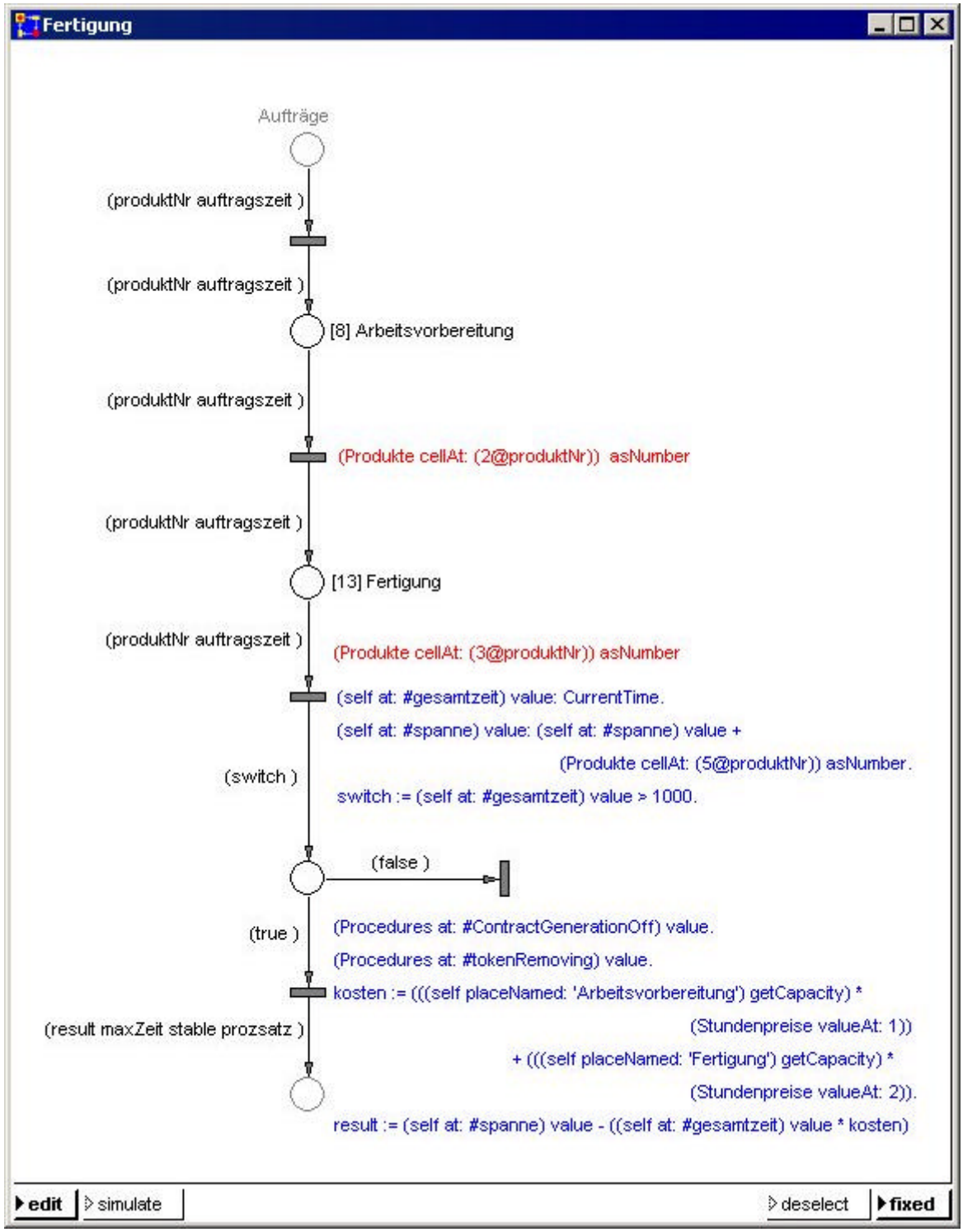
Jede Marke in der Stelle "Auftragserzeugung" trägt zwei Attribute, die Produktnummer und eine Exponentialverteilung, deren Mittelwert die in der Tabelle "'Produkte" angegebene mittlere Zeitspanne zwischen dem Eingang von zwei Aufträgen für das zugeordnete Produkt ist. Diese werden über den Konnektor, welcher der Exponentialverteilung den Namen t zuordnet, zur Transition gebracht. Der nächste Wert der Verteilung "t next" wird als Wartezeit bis zum Schalten der Transition auswertet. Ist diese Zeitspanne vergangen, so wird ein Behälter (der Auftrag) mit zwei Attributen, der Produktnummer "produktNr" und der Startzeit "auftragszeit", erzeugt und in die Stelle "Aufträge" transferiert.



Bemerkenswert ist, dass die Stelle "Aufträge", die als Interface zwischen den Moduln "Auftragserzeugung" und "Fertigung" dient, in den beiden Moduln mit geringerer Inten-

sität als im Modul "Produktherstellung" gezeichnet ist. Dadurch wird kenntlich gemacht, daß die Stelle auf einer höheren Hierarchieebene vereinbart ist.

Das nächste Bild zeigt das etwas vereinfachte Modul "Fertigung".



Die Warteschlangen vor den beiden Abteilungen werden in den Stellen "Arbeitsvorbereitung" und "Fertigung" aufgebaut. In eckigen Klammern ist jeweils die Kapazität der Stelle angegeben, die beim Aufruf der Netzfunktion übergeben wird. Die Wartezeiten für die nachfolgenden Transitionen werden aus der Spalte 2 und 3 der oben gezeigten Tabelle "Produkte" ausgelesen. Das Modul summiert die Einnahmen während einer Zeitspanne von 1000 Stunden auf.

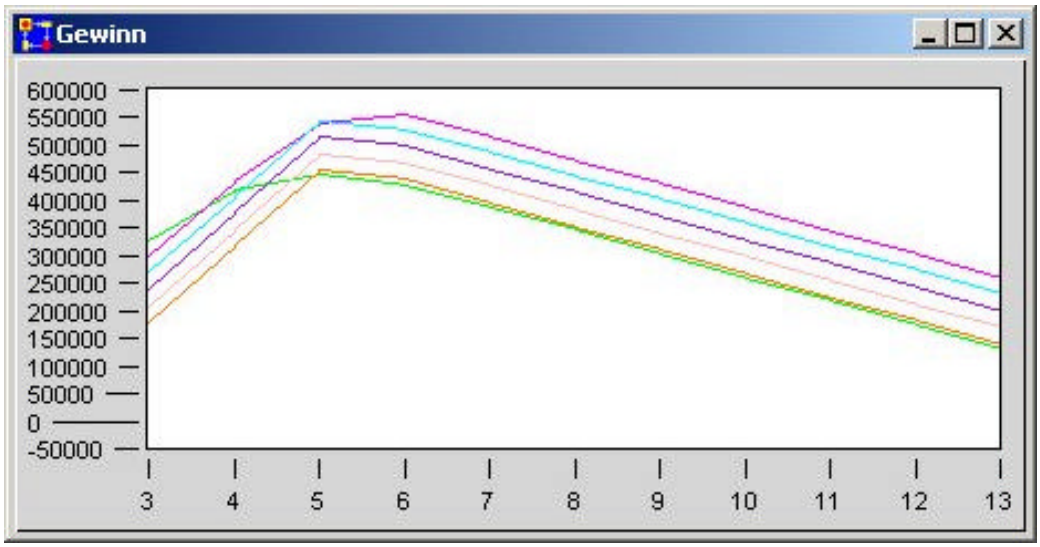
Ist diese Zeit verstrichen, so ergibt sich das Ergebnis "result" als Differenz aus den Einnahmen und den angefallenen Personalkosten.

4.3 Experimentieren mit dem Modell

Die vorangegangene Beschreibung zeigt die wesentlichen Eigenschaften des unter 4.1 angegebenen Geschäftsmodells; das Modell wurde aus Platzgründen etwas vereinfacht dargestellt. Die erweiterte Netzfunktion kann sowohl als normale Netzfunktion als auch mit einem Optimierer eingesetzt werden. Sie ermöglicht außerdem die Vorgabe zusätzlicher Bedingungen, beispielsweise die Vorgabe der maximalen Durchlaufzeit für einen Auftrag. Für die im weiteren beschriebenen Experimente wurde das erweiterte Modell verwendet.

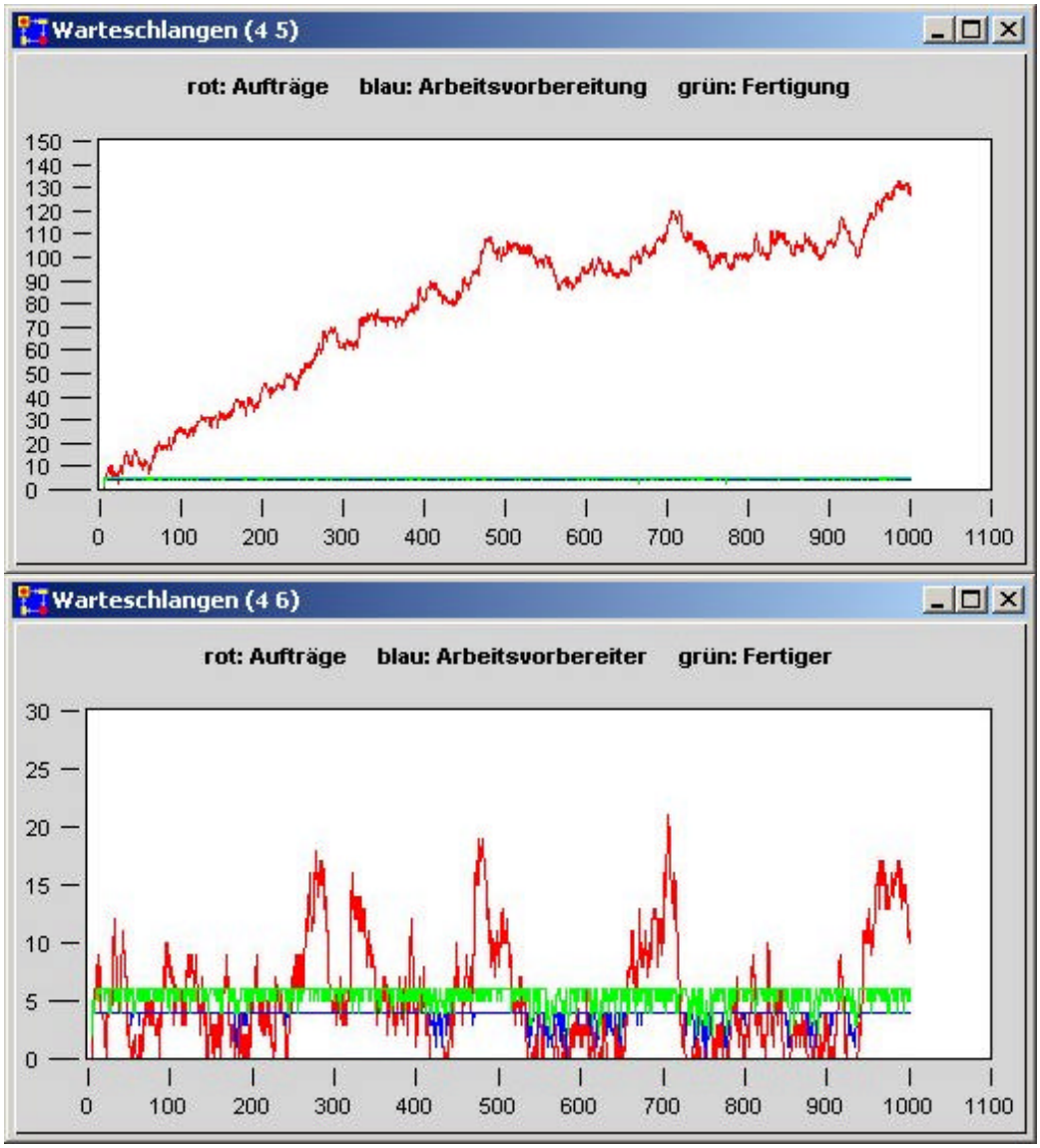
Die wichtigste Frage bei der Auslegung des modellierten Geschäftsprozesses ist die Frage, wie man die Ressourcen wählen muss, damit der Gewinn möglichst groß wird. Aus der früher angegebenen Tabelle "Produkte" gelangt man durch Aufsummieren des mittleren Bedarfs an Facharbeitern in den beiden Abteilungen zu 3.5 bzw. 4.95 Facharbeitern in der Arbeitsvorbereitung bzw. Fertigung. Bei alleiniger Betrachtung der Mittelwerte würde man diese Abteilungen mit 4 bzw. 5 Facharbeitern ausstatten.

Im vorliegenden einfachen Fall von zwei Abteilungen kann die optimale Anzahl von Facharbeitern auch graphisch ermittelt werden. Mit der Netzfunktion wird dabei der Gewinn, der in 1000 Stunden anfällt, in zwei geschachtelten Schleifen berechnet und visualisiert. Das Ergebnis ist in der folgenden Abbildung dargestellt.



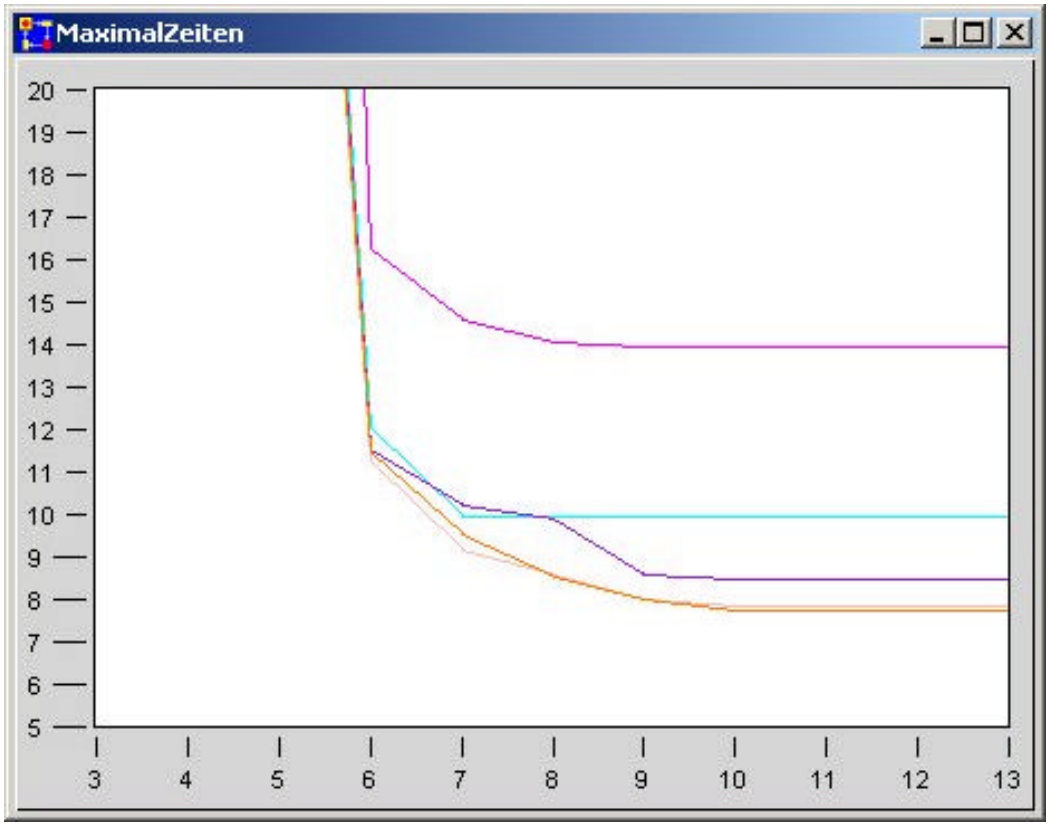
Die Abszisse stellt dabei die Anzahl von Facharbeitern in der Fertigung dar. Die zu einer Kurve gehörige Anzahl von Facharbeitern in der Arbeitsvorbereitung entnimmt man entweder aus der Kurvenfarbe (3:grün,4:magenta,5:zyan,6:purpur,7:rosa;8:orange) oder erhält sie durch Abzählen der Kurven auf der Ordinate von oben nach unten. Ersichtlich liegt das Optimum nicht bei den Mittelwerten (4 5), sondern bei (4 6).

Interessant ist auch das Verhalten der Warteschlangen vor der Arbeitsvorbereitung und der Fertigung. Sie sind zusammen mit der Auftragswarteschlange für die beiden Wertepaare (4 5) und (4 6) in den folgenden beiden Abbildungen dargestellt.

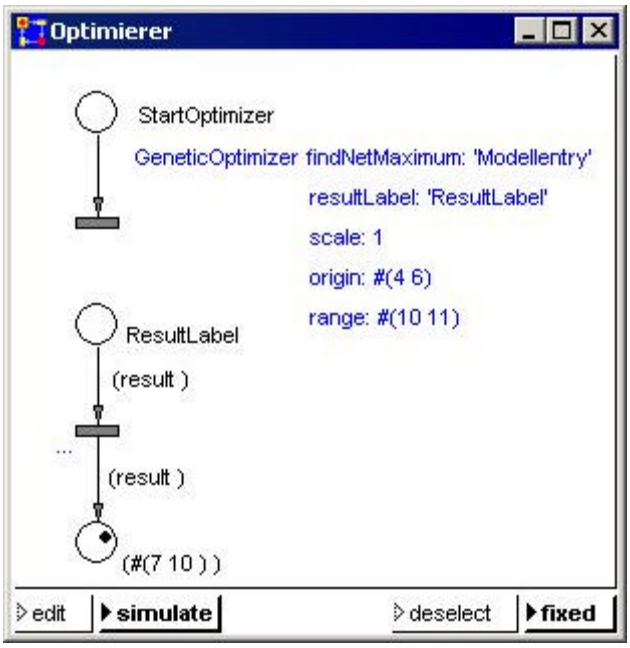


Es zeigt sich, dass die Fertigung bei 5 Fertigern nicht stabil ist, d.h. die Durchlaufzeit eines Auftrags ist nicht unabhängig vom Zeitpunkt der Auftragserteilung. Die Fertigung läuft dagegen stabil, wenn man die Anzahl der Fertiger von 5 auf 6 erhöht.

Diese Zahlen ändern sich, wenn die Auftragsdurchlaufzeit beschränkt wird. Solche Beschränkungen werden häufig bei Produktionen ohne angeschlossenes Lager gefordert. In der folgenden Abbildung sind die maximalen Durchlaufzeiten für Wertepaare im hier relevanten Bereich von 4 bis 8 Arbeitsvorbereitern dargestellt. Die einer Kurve zugeordnete Anzahl von Vorbereitern erhält man über die Kurvenfarben oder durch Abzählen am rechten Rand von oben nach unten.



Wird beispielsweise gefordert, dass die maximale Produktionszeit 8 bzw. 9 Stunden ist, so ergibt sich aus der Abbildung eine Belegung von (7 10) bzw. (7 8).



4.4 Einsatz eines Optimierers

Der Anschluss des in Abschnitt 4.2 beschriebenen Moduls "Produktherstellung" an einen Optimierer zeigt die nebenstehende Abbildung. Verwendet wird hier ein genetischer Optimierer, der das Optimum im Bereich (4 6) bis (10 11) auffinden soll. Dabei sollen nur ganzzahlige Argumente betrachtet werden (scale: 1).

Um die Beschränkung der maximal zugelassenen Durchlaufzeit für Aufträge zu berücksichtigen, wurde die früher angegebene Zielfunktion im Modul "Fertigung" wie folgt geändert:

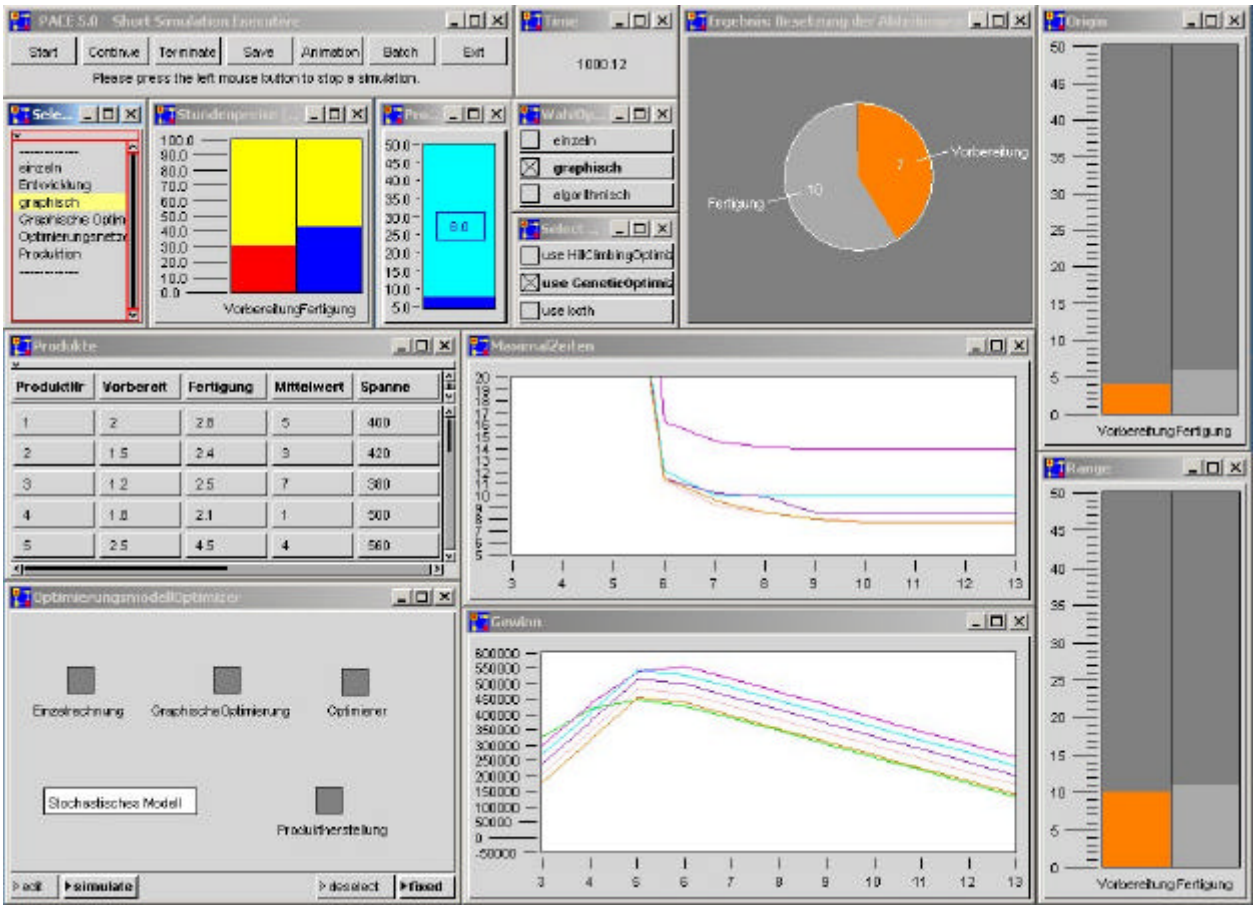
```
result := ((self at: #spanne) value - ((self at: #gesamtzeit) value * einnahmen)) *
                (gemesseneMaxZeit <= geforderteMaxZeit) ifTrue: [1] ifFalse: [0.5]).
```

"gemesseneMaxZeit" ist die während eines Simulationslaufs gefundene maximale Durchlaufzeit eines Auftrags. Das Ergebnis wird halbiert, wenn während der Produktion (im Modell 1000 Stunden) die maximal zugelassene Durchlaufzeit "geforderteMaxZeit" überschritten wird. Für eine maximal zugelassene Durchlaufzeit von 8 bzw. 9 Stunden ergibt sich die schon

aus der visuellen Betrachtung abgeleitete Besetzung der beiden Abteilungen mit 7 Arbeitsvorbereitern und 10 bzw. 8 Fertigern.

4.5 Anwenderoberfläche

Um das Experimentieren mit Modellen zu erleichtern, wurden in PACE zahlreiche graphische und textuelle Ein/Ausgabe- und Visualisierungselemente vorgesehen. Diese lassen sich bequem zu einer Anwenderoberfläche zusammenbauen, die auch von Anwendern, die an der Erstellung eines Modells nicht teilgenommen haben, nach kurzer Einweisung verwendet werden kann. Für die vorliegenden Fallstudie wurde die in der folgenden Abbildung dargestellt Oberfläche verwendet.



Das Modell kann über die "Short Simulation Executive" links oben bedient werden. Zuvor werden über die Balkenschieberegler und die Tabelle die Anfangswerte und weitere früher beschriebene Modellparameter eingegeben. Das Ergebnis wird je nach gewählter Verwendungsart (einzeln, graphisch, algorithmisch) entweder als Torte oder in Form von Kurven ausgegeben.

5. Schluss

Das vorgestellte Modellierungs- und Optimierungsverfahren ist nicht auf den Bereich der Geschäftsprozesse beschränkt, sondern kann für die Modellierung und Optimierung beliebiger diskreter Prozesse eingesetzt werden. Das Optimierungsverfahren wird derzeit schon in der Verkehrstechnik, Bautechnik und in der Logistik eingesetzt. Das Verfahren wurde in PACE 5.0 aufgenommen, um ein systematisches Vorgehen bei der Systemoptimierung zu ermöglichen.

Literatur:

[1] C. Böhnlein: Modellierung des Bullwhip-Effekts mit Hilfe höherer Petri-Netze, wisu, 31 (2002) 8-9, S. 1124-1127

[2] U. Dietel, F. Bennemann: Reihenfolgebildung von Gießaufträgen bei einem sächsischen Metallhersteller, in: M. Rabe, B. Hellingrath: Handlungsanleitung Simulation in Produktion und Logistik – Ein Leitfaden mit Beispielen für kleinere und mittlere Unternehmen, SCS International, 2001, ISBN 1-56555-226-1

[3] U. Dietel, H.-J. Hanisch: Simulation des Bereichs Wärmebehandlung in der Porzellanherstellung, in: M. Rabe, B. Hellingrath: Handlungsanleitung Simulation in Produktion und Logistik – Ein Leitfaden mit Beispielen für kleinere und mittlere Unternehmen, SCS International, 2001, ISBN 1-56555-226-1

[4] S.M.O. Fabricius, E. Badreddin: Stochastic Petri Net Modeling for Availability and Maintainability Analysis, Proceedings of 14th International Congress and Exhibition on Condition Monitoring and Diagnostic Engineering Management (COMADEM), 2001, Manchester, UK

[5] V. Franz: Techniken der Simulation in der Praxis, BFT 5, 2001, S. 20 – 23

[6] B. Eichenauer, K. Scherer: Modellierung und Simulation von intelligenten Gebäudesystemen mit attributierten Petri-Netzen, ikm 16. Internationales Kolloquium über Anwendungen der Informatik und Mathematik in Architektur und Bauwesen, Weimar, Juni 2003

[7] M. Enkelmann: Simulation in der Betonsteinproduktion, BFT 5, 2001, S. 24 - 28

[8] G. Feistl: Findet das Nadelöhr – Simulation von Verpackungsprozessen, neue verpackung 4, 1998, S. 32-34

[9] PACE 5.0 Benutzerhandbuch, IBE Simulation Engineering GmbH, 2002, www.ibepace.com