

TECHNISCHE
CHEMNITZ



UNIVERSITÄT
ZWICKAU

Fakultät Informatik
Lehrstuhl Modellierung und Simulation

Projektarbeit

Thema: Ein Vergleich von Simulations-
software

von

Jens Arnold und Ulf Nieländer

Oktober 1994

Inhaltsverzeichnis

1. Einleitung und Einordnung der vorliegenden Projektarbeit	2
2. Auswahl- und Bewertungskriterien	6
2.1. Neutrale Kriterien	7
2.2. Informatikspezifische Kriterien	7
2.3. Kriterien aus der Sicht der Fabrikplanung	9
3. Das Testmodell "Getriebefertigung"	10
4. An der TU Chemnitz verfügbare und getestete Simulationssoftware	13
4.1. DIOGENES	13
4.2. DOSIMIS-3	19
4.3. PERFECT!	23
4.4. SLAMSYSTEM II	36
4.5. MOSYS	46
4.6. TAYLOR II	50
5. Zusätzlich untersuchte Simulationssoftware	62
5.1. FACTOR/AIM	62
5.2. AutoMod	67
6. Gewichtete Bewertung der Simulationssysteme	71
7. Literatur- und Quellenverzeichnis	72

1. Einleitung und Einordnung der vorliegenden Projektarbeit

Simulation ist eine effektive Methode, in natürlichen oder künstlichen Systemen ablaufende Prozesse zu analysieren, zu beurteilen und zu optimieren. Hat die Simulation ihre Wurzeln in der Physik, Raumfahrt, Energietechnik und Biologie, also dort, wo die analytische Mathematik an ihre Grenzen stieß, so findet sie erst jetzt in Produktion und Logistik ihre angemessene Verbreitung. Einen hohen Stellenwert wird sie dort nicht zuletzt aufgrund der ständig fallenden Hardware-Preise und der durch moderne, ingenieurmäßige Programmiermethoden immer leistungsfähiger werdenden Software erreichen. Simulationswerkzeuge sind nicht mehr Softwaregiganten auf Supercomputern, bedienbar nur durch Spezialisten, sondern für PC's und Workstations verfügbare Anwenderprogramme, die für den Bruchteil der Kosten einer Werkzeugmaschine für jeden Betriebswirt und Ingenieur nutzbar sind.

Auf diese Faktoren hat der Markt mit einem Doppeleffekt reagiert: Einerseits steigt die Nachfrage nach preiswerten, leistungsfähigen Simulationssystemen besonders bei klein- und mittelständischen Unternehmen rasant an, andererseits sind immer mehr kleinere Software-Firmen in der Lage, diese Nachfrage mit eigenen Produkten zu befriedigen. Traditionelle Großanbieter versuchen, durch Downsizing ihre Großrechnerimplementationen ebenfalls in dieser Marktlücke zu plazieren.

Sind diese "betagten" Systeme der Herausforderung durch moderne, objektorientierte Programmierungen gewachsen?

Was kann der Anwender heute von einem Simulationssystem erwarten?

Hält die simulierte Planung der Realität stand?

Oder ist Simulation eine Investition die man sich besser spart?

Wo liegen noch die Grenzen der Simulation?

Eine Beantwortung dieser und ähnlicher Fragen kann immer nur eine Momentaufnahme, ein Snapshot im Prozeß der Marktwirtschaft sein. Der "Marktspiegel Simulationstechnik in Produktion und Logistik" von 1991 [NOCH91] - er war eine große Hilfe bei Fragen des methodischen Herangehens an unsere Aufgabe - ist in seinen Details bereits wieder Makulatur. Wir fanden kein System in der dort beschriebenen Version an der TU Chemnitz-Zwickau vor!

Trotzdem wollen wir versuchen, Antworten auf die gestellten Fragen zu finden, tendenziell - nicht detailliert. Wir wollen nicht zeigen, warum man was wie simulieren kann, sondern womit simuliert werden sollte, damit nach Stunden des Modellierens und Simulierens nicht eine Grafik erscheint, die den Auslastungsgrad einer Maschine nur in 20%-Schritten darstellen kann (so geschehen in PERFECT! !).

Unsere Aufgabenstellung erwuchs aus dem Wunsch des mittelständischen Unternehmens "Heckert-Chemnitzer Werkzeugmaschinen GmbH" in Zusammenarbeit mit dem Lehrstuhl für "Fabrikplanung und Fabrikbetrieb" der TU Chemnitz-Zwickau, ein Simulationstool zur Kundenakquisition im Maschinenbau zu finden, was folgenden Ansprüchen genügen muß:

Schnelle Modellierung und Änderung eines Modells der Größenordnung 1-8 Bearbeitungszentren, 1-4 Spannplätze, ein Palettentransportwagen und 1-7 separate Speicherplätze.

Der Auftragsdurchlauf soll mit dem Ziel simuliert werden, ob die Auslastung aller Komponenten möglichst hoch ist oder zum Beispiel ein Spannplatz, der mit einer zuvor erfolgten statischen Dimensionierung geschaffen wurde, wegfallen kann.

Die Animation soll auf einem hohen Niveau vor Kunden repräsentieren und eventuell auch auf einem Laptop einsetzbar sein.

Das Gesamtsystem (Hard- und Software) muß äußerst kostengünstig sein (unter 10 TDM).

Die Einarbeitungszeit und der laufende Schulungsaufwand muß minimal bleiben.

Eine Realisierung mit dem Simulationssystem TAYLOR II findet der Leser in [ZIMM94].

Für uns stand die Aufgabe, eine Bestandsaufnahme der an der TU Chemnitz-Zwickau verfügbaren Simulationssoftware vorzunehmen. Dabei stellte sich sehr schnell heraus, daß die an der Fakultät für Informatik vorhandenen Simulationssprachen von vornherein aus der Untersuchung herausgenommen werden mußten.

Abb 1.1 zeigt die drei Ebenen auf denen Simulationswerkzeuge angesiedelt werden können.

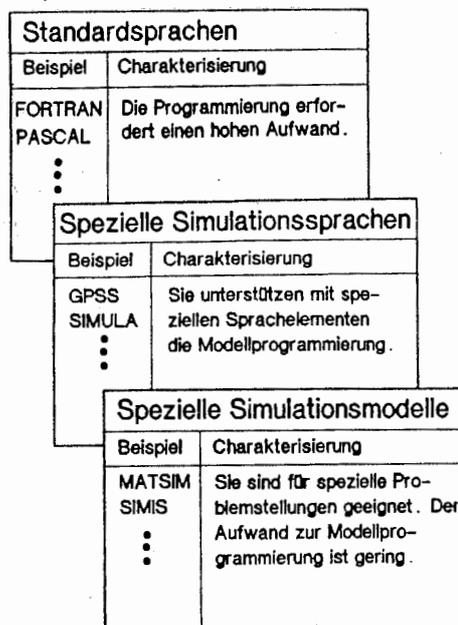


Abb. 1.1 Ebenen der Hilfsmittel zur Erstellung von Simulationsmodellen

Der Modellierungsaufwand auf den Ebenen "Standardsprachen" und "Spezielle Simulationssprachen" steht in keinem sinnvollen Verhältnis zur Modellierung mit speziellen Simulatoren für Produktion, Materialfluß und Logistik. Insbesondere der geringere Abstraktionsgrad auf der höchsten Ebene mit Elementen aus der realen Welt des Materialflusses erhöht hier die Effizienz beträchtlich.

Im Weiteren nahmen wir die an der Fakultät für Maschinenbau und Verfahrenstechnik installierten Simulationssysteme in unsere Bewertung auf:

- DIOGENES
- DOSIMIS-3
- PERFECT!
- SLAMSYSTEM II
- MOSYS
- TAYLOR II

Ihre taxonomische Einordnung in die Welt der Simulationssysteme zeigt Abb. 1.2.

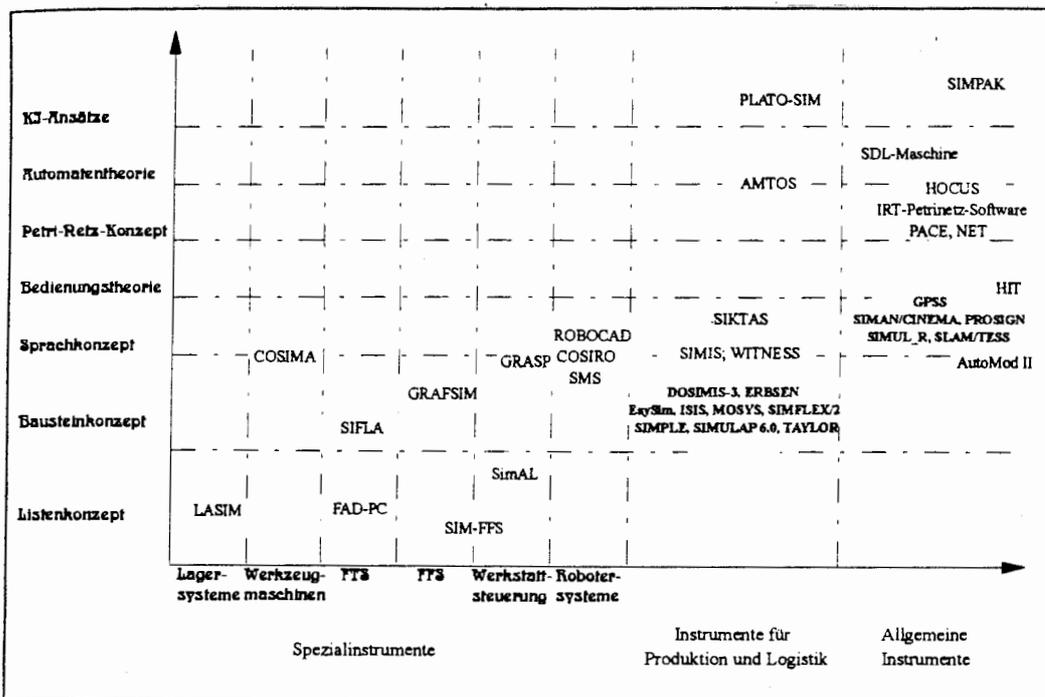


Abb. 1.2 Taxonomie verbreiteter Simulationssysteme [NOCH91]

Im Laufe der Untersuchung kamen dann noch die zwei Systeme FACTOR/AIM und AutoMod hinzu. Bei beiden mußten wir allerdings auf Handbuchrecherchen, Informationsmaterial der Vertreiber und Hinweise von Anwendern zurückgreifen. Sie werden deshalb separat in Kapitel 5 vorgestellt.

Drei der untersuchten Systeme haben wir näher beschrieben:

- * PERFECT! (Kap. 4.3.) als großes System für Workstations;
- * SLAMSYSTEM (Kap. 4.4.) als kleines System für PC's;
- * TAYLOR II (Kap. 4.6.) als Ergebnis des Auswahlprozesses;

Kapitel 6 soll noch einmal die Ergebnisse des Bewertungsverfahrens grob zusammenfassen. Betont werden muß, daß beim Entstehen der vorliegenden Arbeit nicht die Aufgabe "Simulationstool zur Kundenakquisition ..." im Vordergrund stand, sondern eine umfassende Bewertung der an der TU Chemnitz zur Verfügung stehenden Simulationswerkzeuge für Materialfluß und Logistik.

Danksagung

Wir danken Herrn Cand.Ing. Zimmermann für die gute Zusammenarbeit und wünschen ihm viel Erfolg bei der Verteidigung seiner Diplomarbeit. Er hat unsere Sichtweise auf die Simulation aus dem Blickwinkel der Fabrikplanung prägend verändert. Weiterhin danken wir dem Team um Herrn Dr. Fischer vom Lehrstuhl für Fabrikplanung und Fabrikbetrieb, besonders Herrn Dr. Pester, der uns mit dem Betriebssystem VMS vertraut gemacht und mit seiner Erfahrung vor vielen Stolpersteinen im System PERFAC! bewahrt hat. Wir danken Frau Dipl.-Ing. Fischer und den Mitarbeitern des Rechenzentrums Bereich Fabrikplanung und Fabrikautomatisierung für den Zugang zu Hard- und Software sowie für die Handbücher der Simulationssysteme.

Ebenso danken wir Herrn Dr. Lieberwirth vom Rechenzentrum Produktionstechnik. Er hat uns den einzigen Computer mit dem Betriebssystem OS/2 der TU Chemnitz-Zwickau für die Untersuchungen am System FACTOR/AIM zur Verfügung gestellt.

Herrn Lewicki und seinem Team bei der Heckert-Chemnitzer Werkzeugmaschinen GmbH gebührt Dank für die Vermittlung einer Anwendersicht, wie wir sie im Schutze der Universität nie gesehen hätten.

Nicht zuletzt möchten wir uns bei Herrn Prof. Köchel und Frau Gäbel für die freundliche Unterstützung bei der Beschaffung von Informationsmaterial, Handbüchern und Demo-Disketten bedanken.

Chemnitz, den 8. Oktober 1994

Jens Arnold & Ulf Nieländer

2. Auswahl- und Bewertungskriterien

Welche Aspekte eines Simulationssystems sind wichtig und wie hoch ist der Erfüllungsgrad?

Diese Frage steht im Zentrum, wenn es um eine Analyse und Bewertung von Simulationssoftware nach allgemeinen (neutralen), informationstheoretischen und anwenderspezifischen Kriterien geht.

Eine sehr gute Auflistung solcher Kriterien mit ihren Erfüllungsmöglichkeiten findet der interessierte Leser in [NOCH91]. Im folgenden werden wir die für unsere Untersuchungen relevanten neutralen und informatikspezifischen Kriterien mit Beispielen ihrer möglichen Ausprägung aufzählen. Für die anwenderspezifischen Kriterien möchten wir zusätzlich auf [ZIMM94] hinweisen. Dort wird der gezielte Auswahlvorgang für ein Simulationssystem zur Kundenakquisition eines mittelständischen Maschinenbau-Betriebes (Heckert-Chemnitzer Werkzeugmaschinen GmbH) beschrieben.

Im Laufe unserer Untersuchungen hat sich gezeigt, daß alle Simulationssysteme auf einige Grundkonzepte zurückzuführen sind (Abb. 2.1), was einerseits die Beschreibung vereinfacht, andererseits zu begrifflichen Verwirrungen führen kann. Wir versuchen trotzdem eine einheitliche Begriffswelt zu verwenden, obwohl die Hersteller der Simulationsinstrumente die Begriffe teilweise unterschiedlich interpretieren.

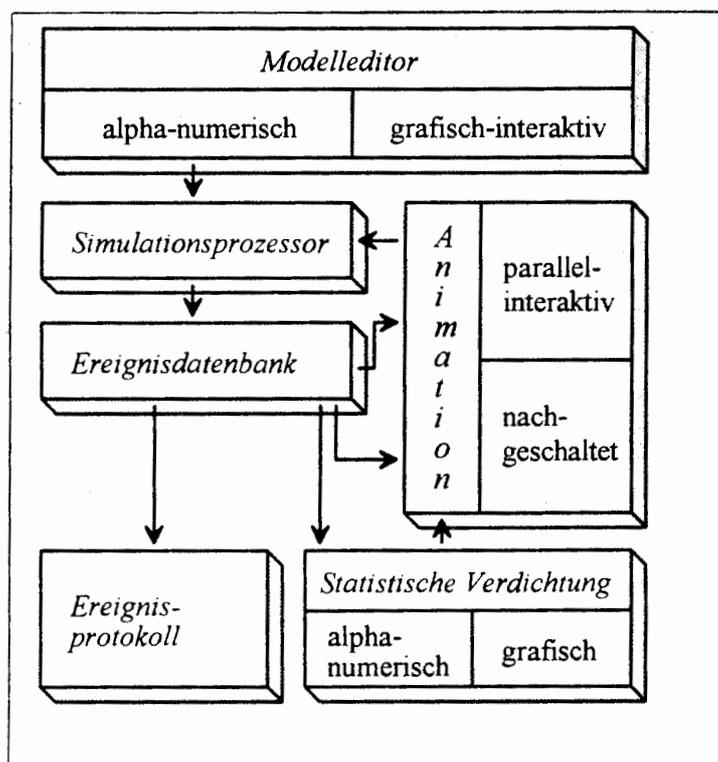


Abb. 2.1 Komponenten einer Simulationssoftware [SPIE91]

2.1. Neutrale Kriterien

Unter dieser Rubrik werden alle die Aspekte betrachtet, welche allgemein für große Softwaresysteme von Bedeutung sind. Sie dienen besonders zur Bewertung der strategischen Ziele, die mit einem Simulationssystem erreicht werden sollen, wie Einsatzzeitraum, Einsatzintensität, Schulungsaufwand etc.

* Marktpräsenz

Verbreitung, Branchen, Unternehmensgröße;

* Ergonomie

Nutzerinterface, Bedienbarkeit, Sprache, Handbuch;

* Erlernbarkeit

Schulungsaufwand, Kenntnisse, Abstraktionsgrad;

* Sicherheitsfunktionen

Kopierschutz, Zugang, Nutzerklassen;

2.2. Informatikspezifische Kriterien

Aus der Sicht der Informatik werden wir hier vor allem programmierungstechnische Aspekte, aus denen sich entscheidende Aussagen über die Leistungsfähigkeit und Funktionalität des Simulationssystems ergeben, untersuchen. Viele hier untersuchte Fragen werden für den Anwender jedoch weniger von Interesse sein, da es sich hierbei um interne Konzepte handelt, die nach außen verdeckt bleiben. Der Simulationsspezialist hingegen wird Antworten finden auf die Frage, warum ein Simulationssystem gerade diese Leistungsmerkmale besitzt und andere dagegen vernachlässigen muß.

* Implementationssprache

C, FORTRAN, PASCAL, Stellung des Simulators, Modularität;

* Modellierung

- Modellsicht:
ereignis-, aktivitäten-, prozeßorientiert;
- Modellabstraktion:
Flußdiagramme bei SLAM II,
Blöcke und Schlangen bei GPSS,
attributierte Entities und Scheduler bei Simscript,
Objekte bei Simula,
Petrie-Netze bei PERFACT und DIOGENES,
Bausteinkonzept bei DOSIMIS, MOSYS, TAYLOR II u.a.;
- Aufbau hierarchischer Modelle;
- Modellbasis (Speicherung von Modellen);
- Modellreplikation;
- Modellerweiterbarkeit;

* Maximale Modellgröße

Anzahl Facilities, Entities, Attribute;

* Statistik

Elementarstatistik, Gesamtstatistik, Anfangswertbelegung, Einschwingphase etc.;

- * Zufallszahlengenerierung
Anzahl generierbarer Verteilungen, Erweiterbarkeit;
- * Robustheit, Verhalten in extremen Situationen
Fehlermeldungen, Systemabstürze, Backups;
- * Eingabe
 - alphanumerisch über Modellspezifikationsprache;
 - grafisch-interaktiv durch Modelleditor 2D/3D;
 - Parametermasken und automatische Fehlererkennung;
 - nutzerdefinierte Datenfiles (Datenstrukturen als Eingabeobjekte);
- * Ausgabe
 - alphanumerisch über Tabellen, Dateien, Reporte;
 - grafisch über Diagramme, Histogramme, Grafiken;
 - 2D/3D-Animation (interaktiv, während und/oder nach der Simulation);
 - Datenstrukturen zur weiteren Verarbeitung durch andere Tools;
 - Druckerunterstützung;
- * Experimente und Debugging
 - Breakpoints, Resetting, Restarting;
 - Checkpoints, Tracing, Monitoring;
 - Snapshots (Speicherauszüge auf Modellebene);
 - Message Observer zwischen Submodellen;
 - Batchbetrieb;

Zwei wesentliche Aspekte wurden von uns nicht berücksichtigt.

- * Das ist zum einen die Performance (Geschwindigkeit der Simulation, Datenaufbereitung und -analyse). Der Grund ist die Hardwareabhängigkeit dieser Größe. Um hier aussagekräftige Werte zu erhalten, müßten alle Simulationssysteme auf gleich konfigurierten Rechnern getestet werden, soweit das bei den einzelnen Systemen überhaupt möglich ist. Die uns zur Verfügung stehende Hardware reichte vom PC i386 bis zur VAX-Workstation. Es muß jedoch eingeschätzt werden, daß eine sinnvolle Simulation bei mittlerer Modellgröße (ca. 30 Facilities) immer noch einen Zeitaufwand von mehreren Stunden bis einige Tage bedeutet.
- * Zum anderen wurden Güteanalysen von Zufallszahlengeneratoren zwar teilweise durchgeführt, aber nicht mit in diese Arbeit aufgenommen. Der Grund liegt einerseits darin begründet, daß es bei manchen Systemen (z.B. PERFACT!) nicht möglich ist, Ausgaben des Generators in eine später auswertbare Datei zu protokollieren und andererseits in der Nutzung von Generatorimplementationen der Programmiersprachen, in denen der jeweilige Simulator geschrieben wurde. Das heißt zum Beispiel, DOSIMIS mit einem C-Compiler auf der VAX/VMS-Workstation übersetzt, nutzt einen anderen Zufallszahlengenerator, als wenn es auf einem PC i386 mit TURBO-C kompiliert wird.

2.3. Kriterien aus der Sicht der Fabrikplanung

Zur Vollständigkeit der Arbeit führen wir diese anwendungsspezifischen Aspekte an dieser Stelle mit an und weisen gleichzeitig darauf hin, daß sie nicht primär Gegenstand unserer Untersuchungen gewesen sind. Wir verweisen wiederum auf [ZIMM94], dem wir auch wörtlich folgen möchten:

1) Bearbeitungsauftrag (Produkt)

- * Eingangsgrößen:
 - Fertigungsmenge (Stück / Fertigung)
 - Bearbeitungsfolge (Vorgänger-Nachfolger-Relationen)
 - Zeitpunkt der Einlastung (statisch / nutzerdefiniert)
- * Ausgangsgrößen:
 - Bearbeitungszeit (Minuten / Schicht)
 - Wartezeit an Einzelstationen (Minuten / Schicht)
 - Durchlaufzeit

2) Transformationselement (Fertigungsplatz)

- * Eingangsgrößen:
 - Einsatzzeit (Minuten / Schicht, Schichten / Tag)
 - Art der Transformation (Bearbeiten, Montieren)
 - Startbedingung (Bereitschaft von Menschen, Material, VWP)
 - Prioritätsregel bei Bearbeitung (FIFO, LIFO usw.)
 - Rüstzeit je Auftrag und Fertigungslos (Minuten / Los)
 - Ausführungszeit je Auftragsart (Minuten / Stück)
 - Fertigungslosgröße (Stück)
 - Verfügbarkeit / Ausfallhäufigkeit (Prozent)
 - durchschnittliche Reparaturzeit (Minuten)
- * Ausgangsgrößen:
 - Ausführungszeit (Minuten / Schicht)
 - Stillstandszeit (Minuten / Schicht)
 - Auslastung

3) Transportelement

- * Eingangsgrößen:
 - Transportlosgröße je Auftragsart (Stück)
 - Einsatzstrecke (Weginformation) je Auftragsart
 - Transportergeschwindigkeit (Meter / Sekunde)
 - Prioritätsregel bei Bearbeitung (FIFO, LIFO usw.)
 - Verfügbarkeit / Ausfallhäufigkeit (Prozent)
 - durchschnittliche Reparaturzeit (Minuten)
- * Ausgangsgrößen:
 - Ausführungszeit (Minuten / Schicht)
 - Stillstandszeit (Minuten / Schicht)
 - Auslastung

4) Pufferelement (Lager)

- * Eingangsgrößen:
 - Pufferkapazität je Auftragsart (Stück)
 - Ein- bzw. Ausgänge des Lagers (Durchsatzleistung)
- * Ausgangsgrößen:
 - Lagerbestand über Zeitraum / Auslastung

5) Hilfselemente (VWP, Kommissionierungsbehälter)

- * Eingangsgrößen:
 - Menge je Auftragsart
 - Einsatzfolge
 - Verfügbarkeit
- * Ausgangsgrößen:
 - Einsatzzeit
 - Wartezeit
 - Auslastung

6) Mensch

- * Eingangsgrößen:
 - Bedienzeit je Auftragsart und Fertigungsplatz
 - Einsatzorte
 - Wegezeit
 - Pausen
 - Prioritätsregel
- * Ausgangsgrößen:
 - Tätigkeitszeit
 - Wartezeit
 - Auslastung

3. Das Testmodell "Getriebefertigung"

Für die gleichartige Bewertung der zu untersuchenden Simulationssysteme entsprechend der im vorangegangenen Kapitel beschriebenen Bewertungskriterien wurde in Zusammenarbeit mit Herrn Cand. Ing. Zimmermann, Lehrstuhl Fabrikplanung und Fabrikbetrieb der TU Chemnitz-Zwickau, ein fiktives Testmodell entwickelt [ZIMM94, Anhang B]. Zum Zeitpunkt der Entwicklung dieses Modells waren die Layout-Daten eines realen Fertigungssystems der Traub Heckert Vertriebs GmbH leider noch nicht für uns verfügbar. Aufbau und Arbeitsweise eines solchen flexiblen Maschinensystems (FMS) sind in [ZIMM94] für das Simulationssystem TAYLOR II beschrieben.

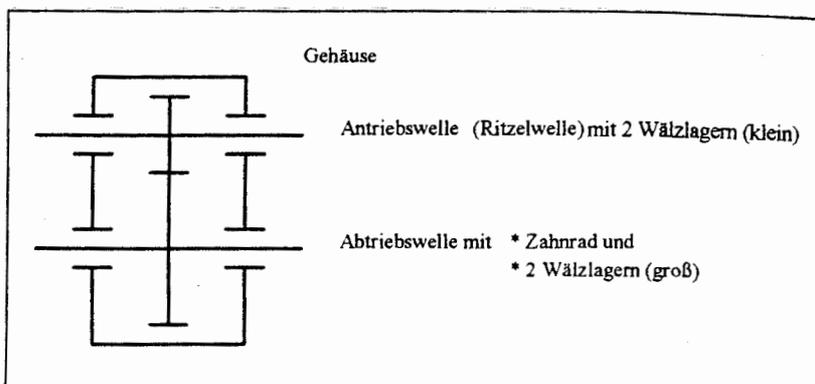


Abb. 3.1 Einstufiges Getriebe

Die Herstellung des in Abb. 3.1 dargestellten Getriebes soll mit der Zielvorgabe simuliert werden, Puffergrößen und Maschinenzahl zu minimieren.

Folgende technologische Annahmen werden dazu getroffen:

(a) Gehäuse und Wälzlager sind Kaufteile und werden vom Eingang direkt zur Montage (5) gebracht.

(b) Fertigung Ritzelwelle:

Eingang	
(1) Drehen Grundform	5 min / Stück
(2) Fräsen Verzahnung	3 min / Stück
(3) Wärmebehandlung	15 min / 10 Stück
(4) Schleifen Verzahnung	1 min / Stück
(1) Außenrundsleifen	0.5 min / Stück
(5) Montage	

(c) Fertigung Zahnrad:

Eingang	
(1) Drehen Grundform	3 min / Stück
(2) Fräsen Verzahnung	4 min / Stück
(3) Wärmebehandlung	15 min / 10 Stück
(1) Innenrundsleifen	0.5 min / Stück
(4) Schleifen Verzahnung	2 min / Stück
(5) Montage	

(d) Fertigung Abtriebswelle:

Eingang	
(1) Drehen Grundform	4 min / Stück
(1) Außenrundsleifen	0.5 min / Stück
(5) Montage	

(e) Fertigung Getriebe:

(5) Montage	7 min / Stück
-------------	---------------

(f) Bedienpersonal:

(1) 1 Werker	0.5 min / Maschine
--------------	--------------------

(g) Transport:

Transportlos = Fertigungslos = 6 Stück

Es werden die Möglichkeiten

- fahrerloses Transportfahrzeug	0.6 m/s
- Stetigförderer (Band)	0.4 m/s

geprüft.

(h) Verfügbarkeit der Anlagen:

- Eingang	0.99
- Speicher	0.98
- Drehmaschine	0.98
- Fräsmaschine	0.92
- Transporter	0.90
- Transportband	0.95

Zum Aufbau des Simulationsmodells wird die in Abb. 3.2 dargestellte Struktur vorgeschlagen. Dabei ist zu jedem Anlagentyp ein Ein-/Ausgabepuffer vorgesehen.

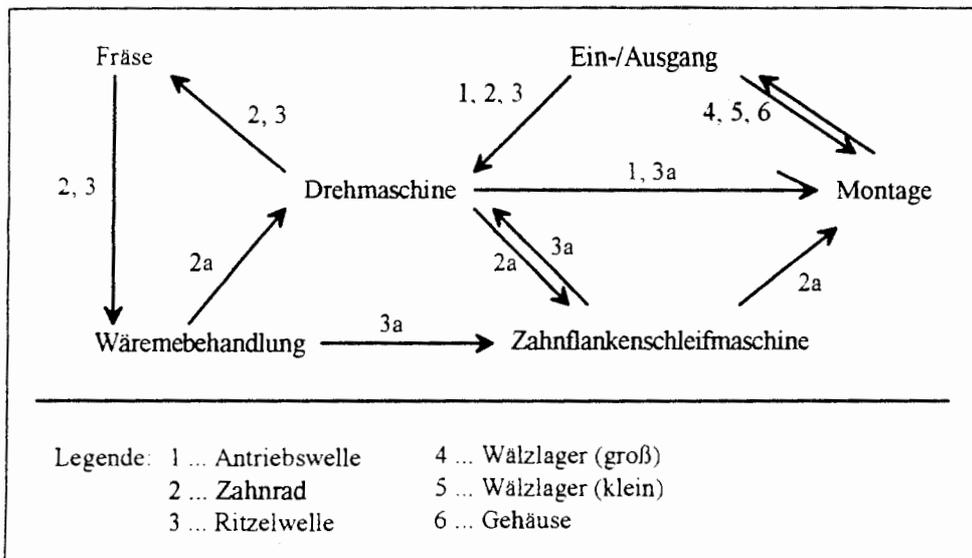


Abb. 3.2 Vorschlag zur Struktur des Modellayouts

4. An der TU Chemnitz verfügbare und getestete Simulationssoftware

4.1. DIOGENES

<i>Hersteller:</i>	ZKI Zentralinstitut für Kybernetik und Informationssysteme, Akademie der Wissenschaften Berlin (abgewickelt) "Dialogorientiertes graphisches Petrie-Netz Entwicklungssystem"
<i>Vertreiber:</i>	GSI Gesellschaft für Steuerungs- und Informationssysteme mbH Rudower Chaussee 5, 12489 Berlin
<i>Preis:</i>	20-50 TDM
<i>Anwendung:</i>	beliebig (Produktion, Logistik, Materialfluß, Rechnernetze, Bundesbahn, ...)
<i>Anwendergruppe:</i>	Simulationsspezialisten
<i>Hardware-Anforderungen:</i>	UNIX-Workstation mit X-Windows, VAX-Workstation mit VMS, PC (i386 oder höher) unter DOS/Windows und OS/2 wird derzeit entwickelt

Neutrale Kriterien

Marktpräsenz:

Das in der ehemaligen DDR konzipierte und in einer neu gegründeten GmbH weiterentwickelte System wird vorrangig in mittelständischen und größeren Unternehmen Deutschlands eingesetzt (Software-Steuerungssystem des FMS1000 Heckert, VLSI-Produktion Numerik/Siemens, Fahrplanoptimierung SABINE Deutsche Bahn AG). Preis und Leistungsvermögen ordnen es in die obere Klasse der Simulationssysteme ein. Sein Einsatz dürfte sich erst ab einer gewissen Größe des zu simulierenden Systems amortisieren.

Ergonomie:

DIOGENES kann je nach Einarbeitungsstand des Nutzers, spezifischer Simulationsaufgabe sowie Hard- und Softwarevoraussetzungen auf verschiedenen Interaktionsebenen betrieben werden. Das MODULA-Programm DGSMAIN übernimmt die grafisch-interaktive Führung des Nutzers durch das gesamte Petriernetzentwicklungssystem. Damit wird der Nutzer durch die Mensch-Maschine-Kommunikation zielgerichtet zu einem gültigen und lauffähigen Simulationsmodell geführt. Eine komfortable Hilfefunktion (Deutsch oder Englisch wählbar) gibt Auskunft über mögliche Kommandos und ihre Schalter sowie über den aktuellen Bearbeitungsstand jedes Netzes. Davon abhängig läßt DGSMAIN jeweils nur bestimmte Kommandos für die weitere Netzbearbeitung zu.

Mit dem Modul NETEVI steht ein grafisch-interaktiver Netzeditor mit voller Fenster- und Mausfunktionalität (auf der Basis von GKS) zur Verfügung. Weiterhin gibt es noch eine Reihe zusätzlicher Module zum Linken, Simulieren, Auswerten etc. von Petrie-Netzen, M-Netzen und Graphen. Die Kommunikationssprache ist durchweg Englisch. Es gibt ein deutschsprachiges Handbuch

Erlernbarkeit:

Der äußert hohe Abstraktionsgrad auf der Ebene von Petrie-Netzen und Graphen setzt fundierte mathematisch-theoretische Kenntnisse über diese Modellierungshilfsmittel voraus. Ist dieses Wissen vorhanden, so ist zur sinnvollen Nutzung des Systems immer noch mit einem relativ hohen Einarbeitungsaufwand zu rechnen.

Sicherheitsfunktionen:

Es bestehen keine Sicherheitsfunktionen gegen Raubkopieren, unberechtigtes Nutzen, Löschen oder Verändern des gesamten Softwaresystems oder einzelner Komponenten und Modelle. Lediglich in UNIX und VMS können die üblichen Zugriffsrechte und Schutzdienste des Betriebssystems genutzt werden.

Informatikspezifische Kriterien

Implementationssprache:

DIOGENES ist in der Programmiersprache PASCAL geschrieben. Der Modul DSMAIN wurde mit MODULA implementiert. Das System ist offen und streng modular aufgebaut.

Modellierung:

Grundlage der Modellierung sind modifizierte, zeitbewertete Petrie-Netze, sogenannte M(odell)-Netze. Diese bestehen i.a. aus der Beschreibung der Netztopologie (Statik, grafische Struktur) und der Beschreibung der Netzsemantik (dynamisches Verhalten). Zur Erleichterung des top-down-Entwurfsprozesses können Transitionen hierarchisch, d.h. in mehreren Ebenen verschachtelt, dargestellt und verfeinert werden. Die Verwendung programmiersprachlicher Elemente des M-Netz-Konzeptes erfordert für jedes Netz einen eigenen Simulator (Parametrierung des Moduls NETSIM), da die die dynamischen Eigenschaften beschreibenden Prozeduren frei programmierbar sind.

Alle Netze und zusätzlichen Beschreibungsmittel werden im Filesystem gespeichert und können somit beliebig geändert, angepaßt und kombiniert werden.

Die Verwendung von Macro- und Subnetzen erlaubt hierarchisch gegliederte Netze, die durch den Modul NETLINK vor der Simulation verknüpft werden müssen.

Maximale Modellgröße:

Die Netzgröße ist theoretisch unbegrenzt. Praktisch limitiert hier nur Speicher- und Rechenzeitbedarf.

Statistik:

In M-Netzen ist die analytische Auswertung durch Methoden der linearen Algebra, der Theorie der abstrakten Sprachen oder der Bedientheorie nur eingeschränkt möglich. Der volle Erreichbarkeitsgraph des Netzes ist nicht zu ermitteln.

Über das gesamte Netz und jeden Knoten und jede Transition kann eine genaue Statistik geführt werden (die in der üblichen Simulationsterminologie üblichen Begriffe stehen in Klammern):

Knotenkenngößen:

- Kapazität (Puffergröße)
- maximale Markierung (maximale Auslastung)
- mittlere Markierung (durchschnittliche Auslastung)

- Frequenz von Markierungs- und Entmarkierungsvorgängen (Durchsatz)
- Markierungsdauer (Bediendauer)
- mittlere Wartezeit

Transitionskenngößen:

- Feueranzahl (Ereignishäufigkeit)
- Aktivitätszeit (Dauer einer Aktivität)

Diese Knoten- und Transitionsattribute werden zu Simulationsbeginn, nach dem Feuern aller gleichzeitigen Transitionen und nach dem Ende der Simulation protokolliert.

Standardcharakteristika von Knoten können an einem Unterbrechungspunkt während der Simulation rückgesetzt werden, um die Einschwingphase aus den Statistiken auszublenden.

Nutzerdefinierte Statistikroutinen können mit Hilfe der PASCAL-ähnlichen NDLG-Prozeduren (Net Description Language for Graphs) hinzugefügt werden.

Zufallszahlengenerierung:

Da die Beschriftung der M-Netze nur mit der interaktiven Netzbeschreibungssprache NDL erfolgt, können und müssen alle Verteilungsfunktionen damit programmiert werden bzw. es kann auf die im jeweiligen PASCAL-Dialekt vorhandenen Zufallsverteilungen zurückgegriffen werden.

Robustheit, Verhalten in extremen Situationen:

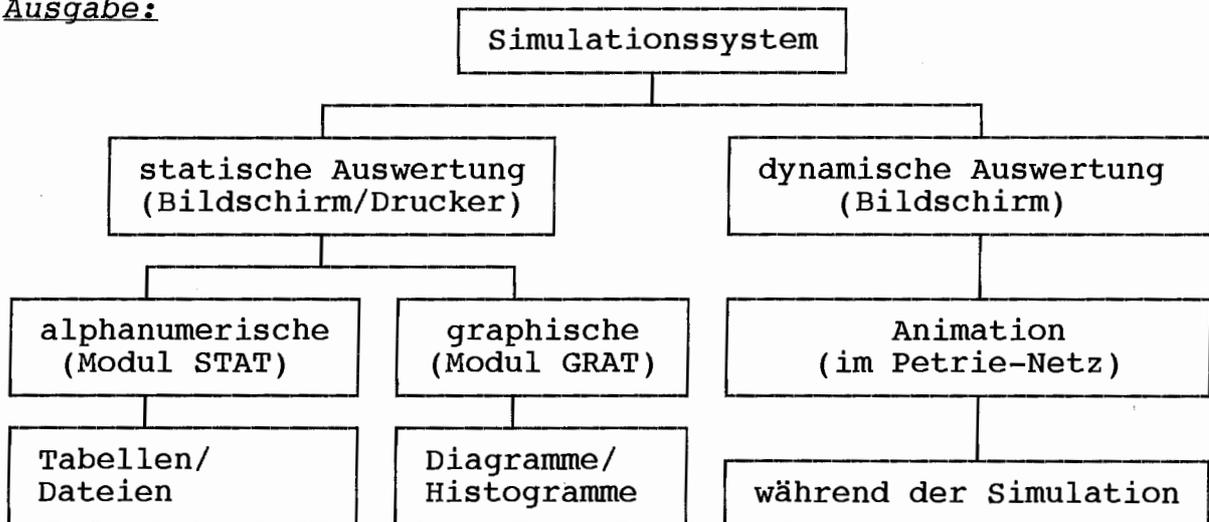
Das gesamte System ist gegenüber Fehlersituationen, Eingabe- und logischen Fehlern relativ sicher. Probleme können sich im Filesystem ergeben, wenn die zu einem Simulationsprojekt gehörenden (15-19) Dateien nicht in einem separaten Verzeichnis abgelegt werden. Weiterhin können die Dateien sehr groß werden, so daß bei Mehrnutzerbetrieb eine entsprechend sorgfältige Ressourcenadministrierung hinsichtlich Hauptspeicher und externem Speicher empfehlenswert ist. undefinierbare Programmabstürze bei unseren Tests sind vermutlich darauf zurückzuführen.

Die Fehlermeldungen sind leider nicht präzise genug und nur über Errorcodes am Ende des Handbuches mit einem Satz dokumentiert.

Eingabe:

Der Modellentwurf kann grafisch-interaktiv mit dem Modul NETEVI durch Aufbau eines Petrie-Netzes oder in Sprachform mit der Netzbeschreibungssprache NDL durch Funktionen und Prozeduren erfolgen. Die Netzdynamik kann nur mittels NDL beschrieben werden. Die Syntax dieser Sprache ähnelt PASCAL, muß aber in ihrer Spezifik erlernt werden. Durch den sehr hohen Abstraktionsgrad muß vor der Eingabe eine genaue Analyse des zu simulierenden Systems erfolgen. Ein korrektes Petrie-Netz kann schwer auf seine Adäquatheit mit dem tatsächlich zu untersuchenden System geprüft werden. Deshalb wurden von der GSI mbH Berlin (Vertreiber) für größere Kunden Eingabesysteme (z.B. SABINE für die Fahrplanoptimierung der Deutschen Bahn AG) geschaffen, die eine einfache, problemorientierte Modellierung gestatten und danach eine automatische Übersetzung des Modells in ein Petrie-Netz vornehmen. Es muß eingeschätzt werden, daß ohne ein solches Hilfsmittel die Eingabe sehr aufwendig ist und sich erst ab einer gewissen Modellgröße überhaupt auszahlt.

Ausgabe:



Die farbige 2D-Animation wird automatisch mit dem Petrie-Netz-Modell erstellt. Mit ihrer Hilfe kann das Petrie-Netz verifiziert und das Entstehen kritischer Situationen (z.B. Deadlock) analysiert werden. Der Nutzer hat die Möglichkeit, selbstdefinierte Teilnetze in die Animation aufzunehmen oder auszuschließen.

Experimente und Debugging:

Für die Beschreibung eines Experimentes können folgende Angaben in Form von NDLG-Anweisungen gemacht werden:

- Initialisierung (Anfangswertbelegung globaler Variablen)
- Abbruchbedingung der Simulation (Vergleichsausdruck)
- Abschluß der Simulation (Ausgabe von Ergebnissen)
- Anzeige von Markeninhalten an Unterbrechungspunkten

Eine Trace-Funktion gestattet nach jedem Feuern einer Transition einen Stop und die Protokollierung des aktuellen Netzzustandes

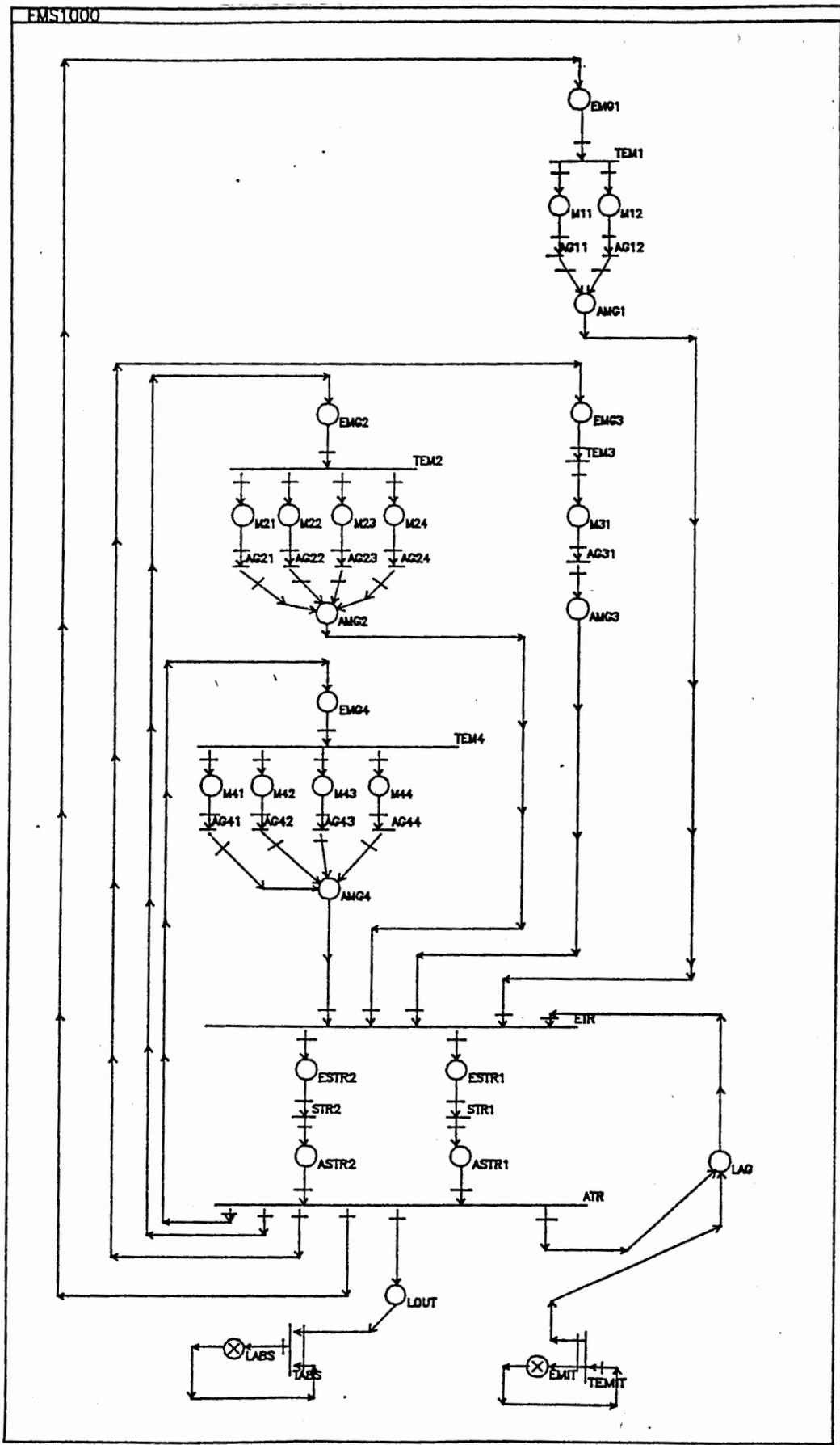


Abb. 4.1 M-Netz-Modell eines flexiblen Fertigungssystems [DIOG92]

Beurteilung des Systems DIOGENES in Hinblick der Nutzung für die Zwecke der Heckert-Chemnitzer Werkzeugmaschinen GmbH

Vorteile:

- hohe Flexibilität, d.h. durch entsprechenden Modellier- und Programmieraufwand an jede beliebige Simulationsaufgabe anpaßbar;
- nutzerdefinierte Daten (z.B. Produktionspläne) einsetzbar;
- Schnittstellen zu anderen Systemen können mit der offenen Programmierschnittstelle in PASCAL realisiert werden;
- es existieren mit dem Software-Steuerungssystem des FMS1000 bereits Erfahrungen in der Firma;

Nachteile:

- äußerst hoher Abstraktionsgrad, dadurch ist der Modellierungsaufwand für Maschinen, Transportsysteme, Lager usw. sehr hoch;
- daraus resultierend ein hoher Einarbeitungsaufwand;
- mindestens eine Person sollte ständig mit dem System arbeiten;
- die 2D-Animation auf diesem Abstraktionsniveau ist für eine ansprechende Präsentation vor Kunden nicht geeignet;
- relativ hoher Preis für Soft- und Hardware;

4.2. DOSIMIS-3

<i>Hersteller:</i>	Universität Dortmund und FhG-IML "Dortmunder Simulation des Materialflusses innerbetrieblicher Systeme"
<i>Vertreiber:</i>	SDZ SimulationsDienstleistungszentrum GmbH Emil-Figge-Straße 76, 44227 Dortmund
<i>Preis:</i>	50-100 TDM
<i>Anwendung:</i>	Produktion, Logistik, Materialfluß
<i>Anwendergruppe:</i>	mit Simulation vertraute Planer und Disponenten
<i>Hardware-Anforderungen:</i>	UNIX-Workstation mit X-Windows, VAX-Workstation mit VMS, PC (i386 oder höher, mind. 4 MB Hauptspeicher, mind. 20 MB freier Festplatten- speicher, VGA-Grafik, DOS mit Windows), Tektron, Apollo, Intergraph

Neutrale Kriterien

Marktpräsenz:

Das in Deutschland entwickelte System wird vorrangig in größeren Unternehmen Europas erfolgreich eingesetzt (Integral, IPL, Industrieplanung, SDZ, Noell). Preis und Leistungsvermögen ordnen es in die obere Klasse der Produktionsplanungs- und Simulationssysteme ein. Sein Einsatz dürfte sich erst ab einer gewissen Größe des zu simulierenden Systems amortisieren.

Ergonomie:

DOSIMIS-3 besitzt eine grafisch-interaktive Benutzeroberfläche mit voller Fenster- und Mausfunktionalität. Damit wird der Nutzer durch die Mensch-Maschine-Kommunikation zielgerichtet zu einem gültigen und lauffähigen Simulationsmodell geführt. Die Kommunikationssprache ist durchweg Deutsch. Es gibt ein deutschsprachiges Handbuch.

Erlernbarkeit:

Durch den eher geringen Abstraktionsgrad ermöglichen die Grundbausteine (Quellen, Senken, Arbeitsstationen, Staustrecken etc.) eine rationale Layoutgestaltung, da sie einerseits der realen Materialflußwelt nachempfunden wurden und andererseits über eine menügeführte Benutzeroberfläche gesteuert werden. Für die Nutzung sind keine informatikspezifischen Kenntnisse notwendig. Somit ist mit einem relativ geringen Einarbeitungsaufwand zu rechnen.

Sicherheitsfunktionen:

Es bestehen keine Sicherheitsfunktionen gegen Raubkopieren, unberechtigtes Nutzen, Löschen oder Verändern des gesamten Softwaresystems oder einzelner Komponenten und Modelle. Lediglich in UNIX und VMS können die üblichen Zugriffsrechte und Schutzdienste des Betriebssystems genutzt werden.

Informatikspezifische Kriterien

Implementationssprache:

DOSIMIS-3 ist in der Programmiersprache C geschrieben. Das System ist offen und modular aufgebaut.

Modellierung:

Ereignisorientierte Modellsicht mit diskreter Zeitfortschreibung. Reichen bei bestimmten Modellen die lokalen Steuerstrategien nicht mehr aus, so stehen dem Anwender globale Entscheidungstabellen zur Verfügung.

Die Modellwelt des Simulators gliedert sich in 6 Ebenen:

- Bausteinebene (26 Bausteintypen);
- Strategieebene: Vorfahrts- und Verteilstrategien;
- Störung- und Pausenebene: zufällig, periodisch, fest;
- Arbeitsbereiche: Teilebearbeitung, Störungsbehebung, etc.;
- Entscheidungstabellen: verallgemeinerte Steuerungsstrategien;
- Programmiererebene: Zugriff auf alle Steuerungs- und Verwaltungsmechanismen

Die Kopplung der Bausteine erfolgt durch Vorgänger-Nachfolger-Relationen. Die Logikbausteine werden durch Petrie-Netze simuliert.

Alle Modelle werden in einer Modelldatenbank gespeichert.

Modellreplikation und -erweiterbarkeit sind möglich.

Es können keine hierarchischen Modelle erstellt werden.

Maximale Modellgröße:

Durch das Bausteinkonzept ist die Modellgröße theoretisch unbegrenzt. Praktisch limitiert hier nur Speicher- und Rechenzeitbedarf. Über das Entscheidungstabellenverfahren sind die Attribute für jeden Entitytyp erweiterbar.

Statistik:

Über das gesamte Modell und jedes Modellelement kann eine genaue Statistik geführt werden:

- Gesamtstatistik, Intervallstatistik, Vorlaufstatistik;
- Zusatzstatistiken (Arbeitsbereiche, Bearbeitungsstationen, etc.);

Selbstdefinierte Statistiken sind nicht erstellbar.

Nutzerdefinierte Daten (z.B. Produktionspläne) können nur über die Programmierschnittstelle (in C) für das System nutzbar gemacht werden.

Zufallszahlengenerierung:

Über den internen Zufallszahlengenerator sind die folgenden Verteilungsfunktionen direkt verfügbar:

- getaktete Verteilung (fest vorgegeben, keine Stochastik)
- Gleichverteilung [a,b]
- Exponentialverteilung
- Normalverteilung
- Erlangverteilung

Weitere Verteilungen können unter Nutzung der Gleichverteilung mit Hilfe der Programmierschnittstelle hinzugefügt werden.

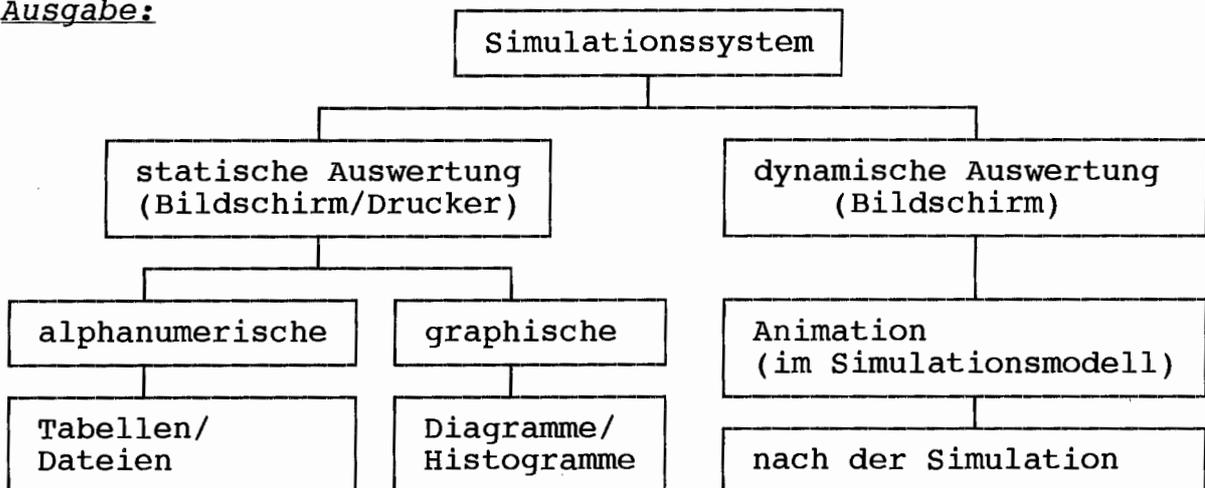
Robustheit, Verhalten in extremen Situationen:

Das gesamte System ist gegenüber Fehlersituationen, Eingabe- und logischen Fehlern sehr sicher. Die Fehlermeldungen sind genügend präzise und verständlich. Im gesamten Testverlauf trat kein undefinierter Zustand, Programmablauf oder -absturz ein.

Eingabe:

Der Modellentwurf erfolgt grafisch-interaktiv durch Aufbau und Parametrierung eines Flußdiagramm-Netzwerkes (26 Bausteine). Dabei können beliebige Modellausschnitte vergrößert werden. Das Setzen der Parameter und Bausteinstrategien geschieht durch das Ausfüllen von Parametermasken oder Entscheidungstabellen. Der Aufbau einer Entscheidungstabelle ist sehr einfach, erfolgt menügesteuert und der Anwender wird so vom System geführt, daß er keinerlei Kenntnis über die Syntax der Bedingungen und Aktionen benötigt. Dadurch wird eine flexible Modellgestaltung ohne Zugriff auf die Programmierschnittstelle ermöglicht.

Ausgabe:



Die 2D-Animation wird automatisch mit dem Modell erstellt. Mit ihrer Hilfe kann das Modell verifiziert und das Entstehen kritischer Situationen (z.B. Deadlock) analysiert werden. Es besteht keine Möglichkeit zur Definition der Animation durch den Nutzer.

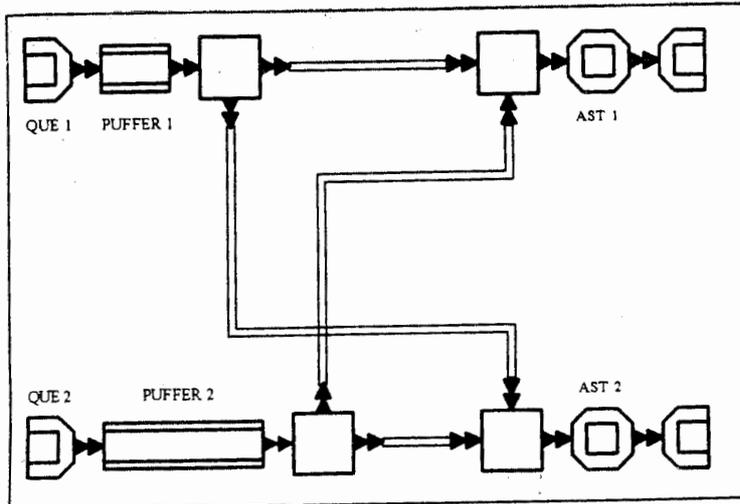


Abb. 4.2 2D-Layout unter DOSIMIS-3 [DOSI93]

Experimente und Debugging:

Im Netzwerk-Modell kann man auf eine Menge von Variablen und Funktionen zugreifen, mit denen der aktuelle Systemzustand beurteilt und verändert werden kann:

- Zufallszahlenverteilungen
- Statistikfunktionen
- Filemanipulation
- Netzwerkstatusauswertung und -manipulation
- Attributmanipulation

Darüber hinaus verfügt das System über einen Konsistenzcheck, der automatisch vor dem Start einer Simulation durchgeführt wird.

Beurteilung des Systems DOSIMIS-3 in Hinblick der Nutzung für die Zwecke der Heckert-Chemnitzer Werkzeugmaschinen GmbH

Vorteile:

- niedriger Abstraktionsgrad, es existieren Bausteine aus der realen Welt des Materialflusses;
- daraus resultierend niedriger Einarbeitungsaufwand;
- hohe Flexibilität, d.h. durch entsprechenden Modellier- und Programmieraufwand an jede Materialflußsimulation anpaßbar;
- nutzerdefinierte Daten (z.B. Produktionspläne) einsetzbar;
- gute und aussagekräftige 2D-Animation;
- Schnittstellen zu CAD-Systemen können mit der offenen Programmierschnittstelle realisiert werden;

Nachteile:

- großes System mit finanziell hohem Aufwand für Soft- und Hardware;
- Eingriffsmöglichkeiten durch Unterbrechung der Simulation sind nur beschränkt möglich;

4.3. PERFECT!

<i>Hersteller:</i>	FhG-IML Fraunhofer-Institut für Materialfluß und Logistik, Bereich II Emil-Figge-Straße 75, 44227 Dortmund
<i>Vertreiber:</i>	SDZ SimulationsDienstleistungszentrum GmbH Emil-Figge-Straße 76, 44227 Dortmund
<i>Preis:</i>	50-100 TDM
<i>Anwendung:</i>	Produktion, Logistik, Materialfluß
<i>Anwendergruppe:</i>	mit Simulation vertraute Planer und Disponenten
<i>Hardware-Anforderungen:</i>	UNIX-Workstation mit X-Windows, VAX-Workstation mit VMS, PC-Variante in Entwicklung

Neutrale Kriterien

Marktpräsenz:

PERFACT! ist die Weiterentwicklung von DOSIMIS und dürfte daher in etwa das gleiche Anwenderspektrum abdecken. Preis und Leistungsvermögen ordnen es in die obere Klasse der Produktionsplanungs- und Simulationssysteme ein. Sein Einsatz dürfte sich erst ab einer gewissen Größe des zu simulierenden Systems amortisieren.

Ergonomie:

PERFACT! besitzt eine grafisch-interaktive Benutzeroberfläche mit voller Fenster und Mausfunktionalität. Damit wird der Nutzer durch die Mensch-Maschine-Kommunikation zielgerichtet zu einem gültigen und lauffähigen Simulationsmodell geführt. Es gibt keine On-Line-Hilfe. Die Kommunikationssprache ist durchweg Deutsch. Es gibt ein deutschsprachiges Handbuch.

Erlernbarkeit:

Durch den eher geringen Abstraktionsgrad ermöglichen die 20 Grundbausteine (Quellen, Senken, Arbeitsstationen, Staustrecken etc.) eine rationale Layoutgestaltung, da sie einerseits der realen Materialflußwelt nachempfunden wurden und andererseits über eine menügeführte Benutzeroberfläche gesteuert werden. Für die Nutzung sind keine informatikspezifischen Kenntnisse notwendig. Somit ist mit einem relativ geringen Einarbeitungsaufwand zu rechnen.

Sicherheitsfunktionen:

Es bestehen keine Sicherheitsfunktionen gegen Raubkopieren, unberechtigtes Nutzen, Löschen oder Verändern des gesamten Softwaresystems oder einzelner Komponenten und Modelle. Lediglich in UNIX und VMS können die üblichen Zugriffsrechte und Schutzdienste des Betriebssystems genutzt werden.

Eine automatische Sicherungsfunktion bei Systemabsturz oder vor Simulationsläufen ist nicht implementiert, so daß der Nutzer vor den Simulationsläufen explizit Sicherungskopien anfertigen sollte.

Informatikspezifische Kriterien

Implementationssprache:

PERFACT! ist in der Programmiersprache C geschrieben. Das System ist geschlossen und nicht erweiterbar.

Modellierung:

Ereignisorientierte Modellsicht mit diskreter Zeitfortschreibung. Dabei spielt in der Modellierungsphilosophie die vertikale Hierarchisierung von Fertigungssystemen eine zentrale Rolle:

- Administration (übergeordnete Planung, gehört nicht zum Fertigungssystem sondern zur Umgebung)
- Disposition (Koordinierung der Subsysteme, Zuordnung von Aufträgen zu Betriebsmitteln und umgekehrt)
- Netzwerke (zu einer organisatorischen Einheit zusammengefügte Prozesse, arbeiten relativ autonom)
- Prozesse (kleinste betrachtete Funktionsbereiche, autonom)

Es existieren auf jeder Ebene (ausgenommen die Administration) Regelsätze und Steuerstrategien.

Reichen bei bestimmten Modellen die Steuerstrategien einer Ebene nicht mehr aus, so stehen dem Anwender globale Entscheidungstabellen zur Verfügung.

Auf der Dispositionsebene können unscharfe Mengen über linguistische Variablen abgebildet werden. Alle Modelle werden in einer Modelldatenbank gespeichert. Modellreplikation und -erweiterbarkeit sind möglich.

Es können hierarchischen Modelle in dem Sinne erstellt werden, daß ein vorhandenes Modell in ein neues Modell als Submodell integriert wird.

Maximale Modellgröße:

Durch das Bausteinkonzept ist die Modellgröße theoretisch unbegrenzt. Praktisch limitiert hier nur Speicher- und Rechenzeitbedarf. Über das Entscheidungstabellenverfahren sind die Attribute für jeden Entitytyp erweiterbar.

Statistik:

Über das gesamte Modell und jedes Modellelement kann eine Statistik geführt werden:

- Bausteinstatistiken
- Objektstatistiken
- Auftragsstatistiken
- Strategiestatistiken
- Kennlinien

Dabei ist die statistische Auswertung durch jede Menge definierbarer Zusatzstatistiken nahezu unbegrenzt. Allerdings ist die Genauigkeit der Auswertung nicht beeinflussbar und leider oft nicht fein genug (z.B. Auslastung von Bearbeitungsstationen sind nur in den Schritten 0, 20, 40, 60, 80, 100 % möglich). Die Einschwingphase muß der Nutzer nach eigenen Kriterien ermitteln. Er kann diese dann mit der Definition von Auswertungszeiträumen aus der Statistik ausblenden. Nutzerdefinierte Daten (z.B. Fertigungsprogramme, Arbeitspläne) müssen in einem definierten Dateiformat vorliegen und können dann vom System eingelesen werden.

Zufallszahlengenerierung:

Über den internen Zufallszahlengenerator sind die folgenden Verteilungsfunktionen direkt verfügbar:

- getaktete Verteilung (fest vorgegeben, keine Stochastik)
- Gleichverteilung [a,b]
- Exponentialverteilung
- Normalverteilung
- Erlangverteilung
- Histogrammverteilung (nur in Quellen und Senken)

Dabei wird im Handbuch besonders darauf hingewiesen, dass alle Verteilungsfunktionen (außer der Histogrammverteilung) mathematisch exakt sind! Weitere Verteilungsfunktionen können nicht hinzugefügt werden.

Robustheit, Verhalten in extremen Situationen:

Systemfehler können auftreten, wenn Parametermasken eigentlich dokumentierte Parameter für bestimmte Funktionen nicht zulassen oder aber Parameter verlangen, die für diese Funktion nicht vorgesehen sind. Zum Beispiel werden beim Verlassen der Lagerbaustein-Maske die Parameter nicht gespeichert oder beim Aufruf einer Parametermaske für eine Verteilung die Parameter für eine andere Verteilung verlangt.

Die Fehlermeldungen sind genügend präzise und verständlich. Insgesamt hinterließ PERFECT! auf diesem Gebiet den typischen Eindruck eines noch in Entwicklung befindlichen Systems.

Eingabe:

Der Modellentwurf erfolgt grafisch-interaktiv durch Aufbau und Parametrierung eines Flußdiagramm-Netzwerkes (20 Bausteine, 4 bewegliche Objekte, 3 Strategiebausteine, Verbindungen). Dabei können beliebige Modellausschnitte vergrößert werden. Die Symbolik ist logisch, aber nicht änderbar. Das Setzen der Parameter und Bausteinstrategien geschieht durch das Ausfüllen von Parametermasken oder Entscheidungstabellen. Der Aufbau einer Entscheidungstabelle ist einfach, erfolgt menügesteuert und der Anwender wird so vom System geführt, daß er keinerlei Kenntnis über die Syntax der Bedingungen und Aktionen benötigt.

Für den Änderungsdienst stehen die Funktionen "Kopieren", "Parameter", "Verschieben" und "Löschen" zur Verfügung.

Die nächste Seite zeigt das 2D-Layout für das Beispiel "Getriebefertigung" mit PERFECT!.

Physische Modell- elemente:



Quelle



Senke



Staustrecke / Puffer



Foederstrecke



Verteilelement



Zusammenfuehrelement



Beladestation



Entladestation



Bearbeitungsstation



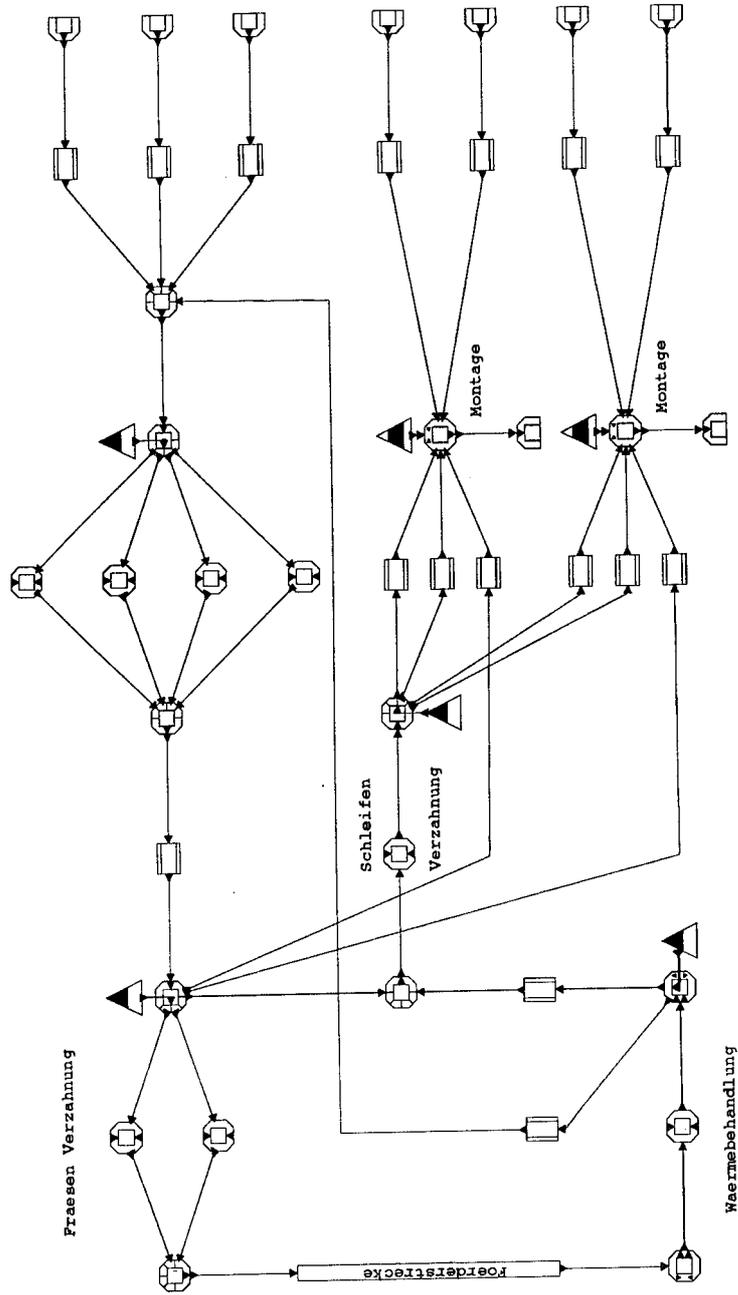
Dispositionsbaustein



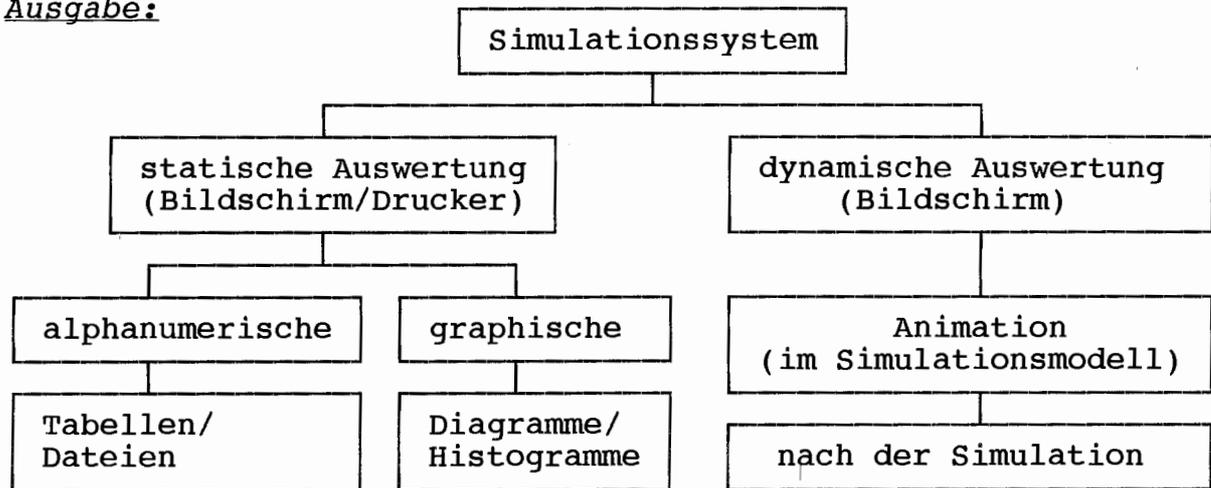
Netzwerkbaustein

Logische Modell- elemente:

Drehen Grundform
I/A-Rundschleifen



Ausgabe:



Die 2D-Animation wird automatisch mit dem Modell erstellt. Mit ihrer Hilfe kann das Modell verifiziert und das Entstehen kritischer Situationen (z.B. Deadlock) analysiert werden. Es besteht keine Möglichkeit zur Definition der Animation durch den Nutzer.

Experimente und Debugging:

Vor einem Experiment kann das System mit einem Anfangszustand initialisiert werden. Das kann auch der Endzustand eines zuvor abgebrochenen Simulationslaufes sein. Darüber hinaus verfügt das System über einen Konsistenzcheck. Über Statistik- und Tracefunktionen kann für das ganze Modell oder einzelne Bausteine ein Bewegungsprotokoll geschrieben werden. Die Simulation kann dabei aber nicht angehalten, sondern nur abgebrochen werden.

Der gesamte Systemzustand kann zu einem definierbaren Zeitpunkt in eine Datei geschrieben werden (Snapshot). Während der Simulation können Systemzustände wie Bestände von Bausteinen und Zugangs-/Abgangsdiagramme angezeigt werden (Monitoring).

PERFACT! erlaubt keinen Batchbetrieb, so daß für jeden Simulationslauf ein Bediener anwesend sein muß. Die Fehlersuche gestaltet sich durch die Off-Line-Animation und das Fehlen spezieller Abfragefunktionen für Systemzustände schwierig.

Beurteilung des Systems PERFACT! in Hinblick der Nutzung für die Zwecke der Heckert-Chemnitzer Werkzeugmaschinen GmbH

Vorteile:

- niedriger Abstraktionsgrad, es existieren Bausteine aus der realen Welt des Materialflusses;
- daraus resultierend niedriger Einarbeitungsaufwand;
- hohe Flexibilität, d.h. durch entsprechenden Modellierungsaufwand an jede Materialflußsimulation anpaßbar;
- nutzerdefinierte Daten (z.B. Arbeitspläne) einsetzbar;
- gute, aussagekräftige und schnelle 2D-Animation;

Nachteile:

- großes System mit finanziell hohem Aufwand für Soft- und Hardware;
- eingeschränkte Fehlersuche durch nur Off-Line-Animation;
- keine dynamische Darstellung des Produktwandels (Symbolik)

Ausführlichere Beschreibung

Bei der Übertragung einer Fabrik in ein Fabrikmodell wird für jedes Fabrikelement, das für die Simulation relevant ist, ein abstraktes Gegenstück im Rechner benötigt. Bei PERFAC! werden einerseits die physischen, d.h. "anfaßbaren" Fabrikelemente, wie beispielsweise Maschinen und Werkstücke, und andererseits die logischen Fabrikelemente, wie beispielsweise Arbeitspläne und Strategien, betrachtet.

Modellelemente zur Abbildung der Physis:

Die mobilen physischen Elemente der Fabrik heißen Objekte und werden klassifiziert in Werkstücke, Fertigungshilfsmittel, Transporthilfsmittel und Transportmittel. Zu jeder Objektklasse sind Parameter angebar. Allgemeine Parameter sind dabei Länge und Breite sowie Kapazität (fest, längenabhängig, typabhängig) des Objektes. Der letztgenannte Parameter beschreibt, ob ein Objekt (Träger) andere Objekte (Lasten) tragen darf.

- Die Werkstücke sind dabei die Objekte im System, die das eigentliche zu fertigende Produkt darstellen. Hierbei kann auch der werkstückspezifische Parameter Arbeitsplan angegeben werden.
- Fertigungshilfsmittel sind die Objekte im System, die bei der Bearbeitung der Werkstücke zusätzlich zu den Maschinen benötigt werden, z.B. Werkzeuge. Fertigungshilfsmittelspezifische Parameter sind Menge, Einheiten und Standzeit. Sie sind sinnvoll insbesondere dann, wenn Fertigungshilfsmittel verbraucht bzw. benutzt werden und diese bei Verbrauch bzw. Verschleiß ersetzt werden müssen.
- Transporthilfsmittel sind die Objekte im System, die zum Transport anderer Objekte dienen, aber über keinen eigenen Antrieb verfügen, z.B. Paletten.
- Transportmittel sind die Objekte im System, die zum Transport anderer Objekte dienen und über einen eigenen Antrieb verfügen, z.B. Gabelstapler. Als transportmittelspezifische Parameter können Geschwindigkeit, Ladung (aktuell, kritisch, maximal) und Verbrauch (bei Leerfahrt, Lastfahrt, Stillstand, Übergabe) angegeben werden.

Die stationären physischen Objekte der Fabrik werden mit sogenannten Bausteinen abgebildet. Sie werden grob klassifiziert in Systemschnittstellen, Bearbeitung, Transport und Lager. Eingeordnet in diese vier Klassen existieren 20 Bausteinarten, die als graphische Symbole direkt im Layout positioniert und über Ihre Eingänge und Ausgänge miteinander verbunden werden können. Jeder Baustein wird anschließend mit spezifischen Werten zu Geschwindigkeit, Länge, Bearbeitungszeit usw. parametrisiert. Weiterhin ist zu jedem Baustein ein typisches Simulationsverhalten hinterlegt, das darüberhinaus über eine Schnittstelle zu Strategien beeinflusst werden kann. Die meisten Bausteine können initialisiert werden.

- Quellen sind Eingangsschnittstellen des Modells. Darin können Objekte gemäß Fertigungsprogramm (externe Datei) oder statistischer Verteilung generiert werden und an das System abgegeben werden. Quellen haben unendliche Kapazität.
- Senken sind die Ausgangsschnittstellen des Modells. Hier werden in bestimmten Takten Objekte gelöscht. Senken haben die Kapazität 1.
- Eine Förderstrecke dient zur Abbildung von Stetigfördersystemen, bei denen Objekte nicht aufeinander auffahren können, z.B. Bandförderer. Solche Förderstrecken haben längenabhängige Kapazität.
- Eine Stau­strecke ist ein Förderbaustein, bei dem die Objekte aufeinander auffahren können, z.B. Rollenförderer mit mehreren Antrieben. Auch Stau­strecken haben längenabhängige Kapazität.
- Der LIFO-Puffer dient zur Abbildung von Stetigfördersystemen, bei denen die Objekte aufeinander auffahren können. Dabei fährt das zuletzt eingefahrene Objekt als erstes wieder aus. LIFO-Puffer haben ebenfalls längenabhängige Kapazität.
- Verteilelemente dienen zum Abbilden einer speziellen Kreuzung. Sie besitzen die Kapazität 1 und benötigen immer eine Verteilstrategie, die entscheidet, durch welchen Ausgang das im Verteilelement befindliche Objekt ausfahren soll.
- Der Ausschleuser-Baustein ist eine spezielle Form des Verteilelementes mit zwei Ausgängen. Der Hauptweg verläuft vom Eingang zum ersten Ausgang. Bei Bedarf werden einzelne Objekte über den Ausschleusweg abgezweigt.
- Zusammenführelemente dienen zum Abbilden einer speziellen Kreuzung. Sie besitzen die Kapazität 1 und benötigen immer eine Vorfahrtsstrategie, die entscheidet, welches von mehreren Objekten an verschiedenen Eingängen des Bausteins einfahren soll.
- Der Einschleuser-Baustein ist eine spezielle Form des Zusammenführelementes mit zwei Eingängen. Der Hauptweg verläuft vom ersten Eingang zum Ausgang. Bei Bedarf werden einzelne Objekte über den Einschleusweg zugeführt.
- Kreuzungen sind Zusammenführelement und Verteilelement gleichzeitig. Sie besitzen die Kapazität 1 und benötigen immer eine Vorfahrts- und eine Verteilstrategie. Damit wird geregelt, von welchem Eingang ein Objekt einfahren darf und an welchem Ausgang es ausfahren soll.
- Die Arbeitsstation wird in PERFAC! verwendet, um Auftragsbearbeitungen durchzuführen. Hierzu gehören das physische Verändern von Objekten (insbesondere Werkstücken) sowie andere (logische) Vorgänge wie z.B. ein Prüfvorgang oder eine Qualitätsfreigabe. Generell können hier alle Vorgänge modelliert werden, die im weitesten Sinne zur Fertigungstechnik gehören. Wesentliche Charakteristik dieses Bausteins ist dabei, daß objektspezifische Zeiten (für Umrüstung, Bearbeitung) verbraucht werden, bevor das jeweilige Objekt die Arbeitsstation wieder

verlassen kann. Eine Arbeitsstation hat die Kapazität 1 (Grundobjekt). Zusätzlich zu diesem Grundobjekt (inkl. beliebig komplexer Beladungsliste) können beliebig viele Fertigungshilfsmittel in den Baustein einfahren, sofern diese für eine Bearbeitung an der jeweiligen Maschine vorgesehen sind.

Es gibt zwei Möglichkeiten der Parametrierung einer Bearbeitungsstation. Einerseits können die für den Fertigungsprozeß notwendigen Informationen in Form von Arbeitsplänen vorliegen. Ein Arbeitsplan dient zur Strukturierung der Arbeitsaufgabe sowie zur Arbeitsverteilung und Arbeitssteuerung. Hierzu enthält er alle Arbeitsgänge, die zu Fertigung eines Produktes notwendig sind. Es wird die Reihenfolge der Arbeitsgänge mit Verweisen auf die entsprechenden Bearbeitungsorte gespeichert. Auch werden Verweise auf die benötigten Fertigungshilfsmittel geführt. Außerdem werden Rüst- und Bearbeitungszeiten pro Arbeitsgang erfaßt.

Liegen andererseits die Informationen nicht oder nur teilweise in Form von Arbeitsplänen vor, so werden die für den Fertigungsprozeß notwendigen bzw. fehlenden Daten explizit über mehrere Masken an der Bearbeitungsstation parametriert.

- Montageelemente bilden einen Montagevorgang ab. Auf ein Grundobjekt werden mehrere Objekte montiert. Die Objekte fahren dann als eine Einheit weiter durch das System. Montageelemente haben die Kapazität 1 (Grundobjekt) und benötigen immer die Zustandsstrategie Montage, in der die Art und Weise vorgegeben ist, wie die zu montierenden Objekte aufgenommen und verarbeitet werden.

- Eine Beladestation bildet einen Beladevorgang ab. Auf ein Grundobjekt werden mehrere Objekte aufgeladen und können später wieder entladen werden. Beladestationen haben die Kapazität 1 (Grundobjekt) und benötigen immer die Zustandsstrategie Beladen, in der die Art und Weise vorgegeben ist, mit der die aufzuladenden Objekte aufgenommen und verarbeitet werden.

- Demontageelemente bilden einen Demontagevorgang ab. Aus einem Grundobjekt werden mehrere Objekte neu generiert. Demontageelemente haben die Kapazität 1 (Grundobjekt) und benötigen immer die Zustandsstrategie Demontage und eine Verteilstrategie, in der die Art und Weise vorgegeben sind, mit der das Grundobjekt demontiert wird, d.h. neue Objekte generiert und abgegeben werden.

- Eine Entladestation bildet einen Entladevorgang ab. Von einem Grundobjekt werden mehrere Objekte entladen. Diese Objekte müssen vorher auf das Grundobjekt aufgeladen worden sein. Entladestationen haben die Kapazität 1 (Grundobjekt) und benötigen immer die Zustandsstrategie Entladen, in der die Art und Weise vorgegeben ist, mit der die zu entladenden Objekte abgeladen und abgegeben werden.

- Der Drehtisch-Baustein bildet eine Förderung ab, die aus einer Drehung besteht. Er hat die Kapazität 1 und benötigt bei mehr als einem Ein- und Ausgang immer eine Vorfahrts- und eine Verteilstrategie, die das Einfahren in diesen Baustein und das Ausfahren aus diesem Baustein regeln.
- Ein Verteilwagen kann in zwei Dimensionen verfahren. Er hat die Kapazität 1 und benötigt bei mehr als einem Ein- und Ausgang immer eine Vorfahrts- und eine Verteilstrategie, die das Einfahren in diesen Baustein und das Ausfahren aus diesem Baustein regeln. Die Positionen der Ein- und Ausgänge müssen angegeben werden zur Bestimmung der Wege in X- und Y-Richtung.
- Der Förderkreis-Baustein bildet eine kreisförmige Förderung ab. Er bewegt sich getaktet weiter, wobei der Abstand der Objekte fest ist. Die Kapazität ist als Parameter einzugeben. Bei mehr als einem Ausgang ist immer eine Verteilstrategie erforderlich, die entscheidet, ob ein Objekt im Förderkreis an dem jeweiligen Ausgang ausfahren darf.
- Der Lager-Baustein modelliert ein Lager mit mehreren Lagerbereichen (Boxen). Die Kapazitäten der Lagerbereiche sind als Parameter einzugeben. Dieser Baustein benötigt immer eine Entscheidungstabellestrategie.
- Schließlich existiert noch ein Grob-Baustein mit zwei Anwendungsgebieten. Einerseits wird er bei der Grobplanung benutzt, wenn man einen Fertigungsbereich nur durch sein Gesamtzeitverhalten abbilden will. Andererseits kann man einen Fertigungsbereich, den man schon fein simuliert hat, automatisch in einen Grob-Baustein umwandeln.

Die Leistung eines gesamten Fabriksystems hängt u.a. von der Verfügbarkeit der Systemelemente ab. Störungen und Pausen beschränken diese in deterministischer oder stochastischer Weise. Sie sind durch zwei wesentliche Merkmale gekennzeichnet: Zum einen durch den Zeitpunkt und die Dauer der Störung oder Pause, und zum anderen durch die von der Störung oder Pause betroffenen Bausteine. Die der betreffenden Störung bzw. Pause zugeordneten Bausteine sind i.allg. Materialflußbausteine.

Störungen und Pausen werden in der gleichen Weise modelliert.

- Für feste Störungen/Pausen müssen die Startzeitpunkte und Dauer der Störungen bzw. Pausen angegeben werden.
- Zufällige Störungen/Pausen werden durch Eingabe der Dauer einer Störung (Störzeit) bzw. Pause (Pausenzeit) und des Zeitabschnittes zwischen zwei Störungen (Funktionszeit) bzw. Pausen (Arbeitszeit) modelliert. Dabei werden die bekannten Verteilungsfunktionen verwendet.
- Periodische Störungen/Pausen erfordern die Eingabe des Starts der ersten Periode und der Periodendauer. Darüber hinaus müssen die Stör- bzw. Pausenereignisse mit Startzeitpunkt und Dauer innerhalb der Perioden festgelegt werden.

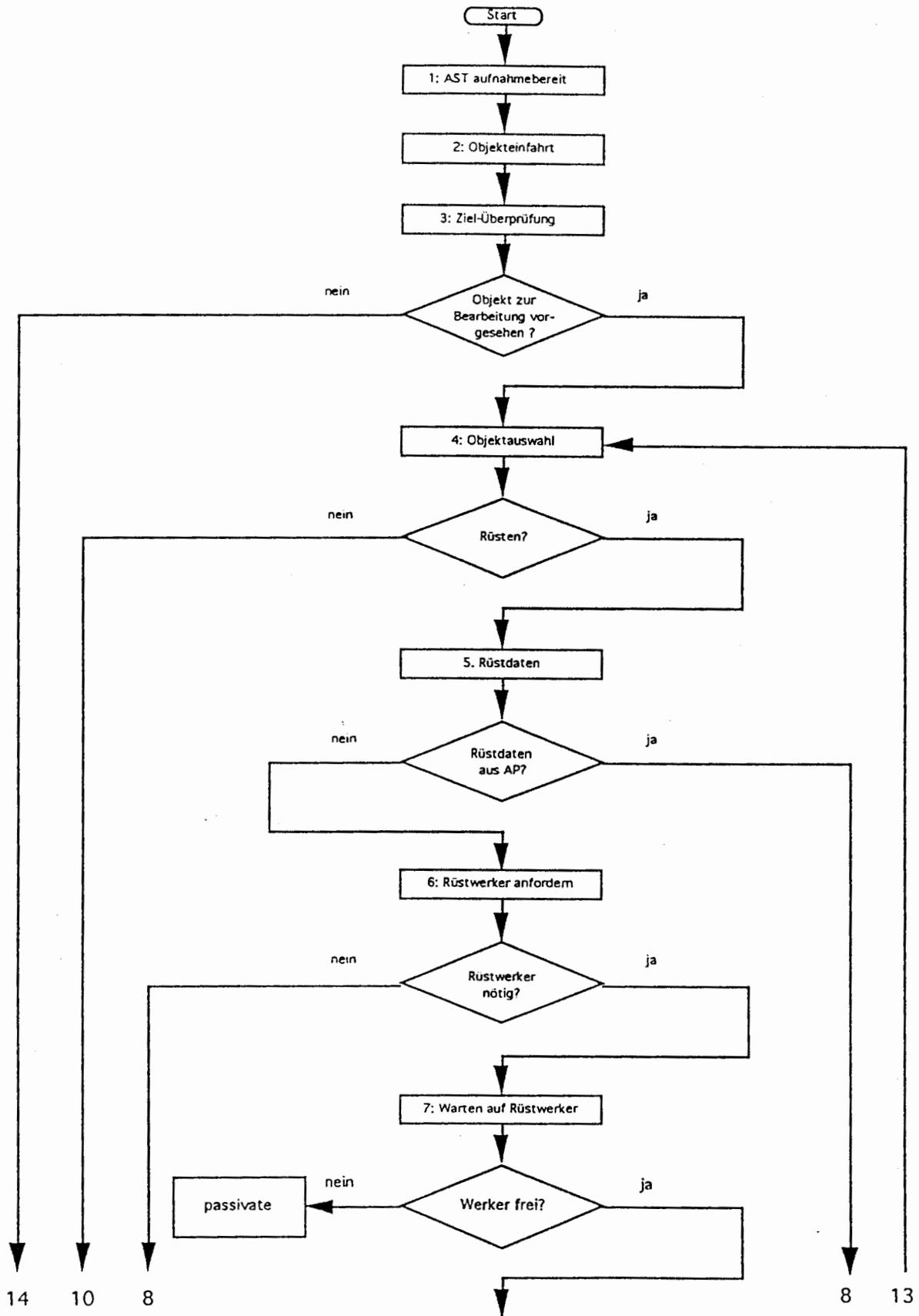


Abb. 4.3 (a) Flußdiagramm über den zeitlichen Ablauf der Zustände in einer Arbeitsstation aus [PERF93]

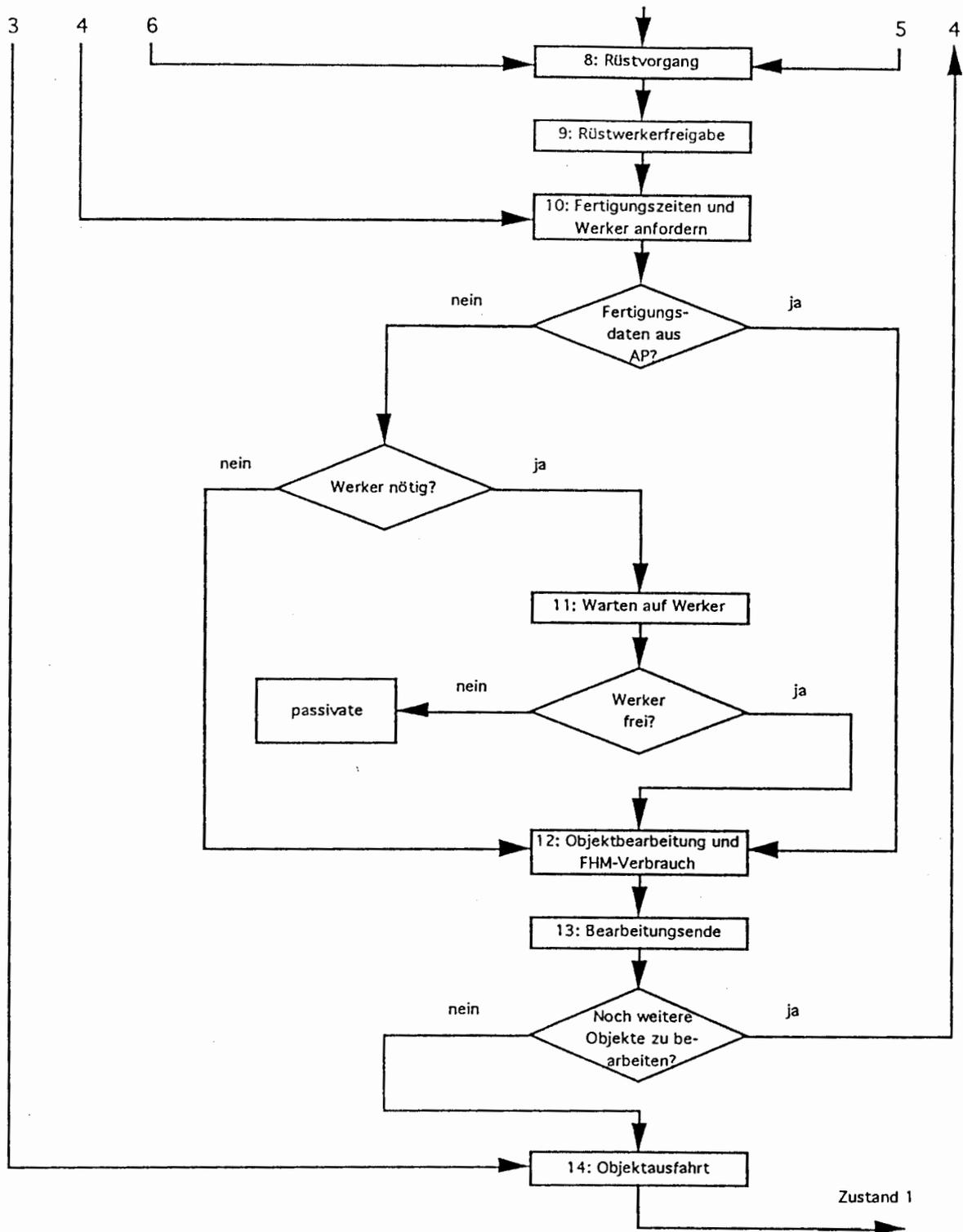


Abb.4.3 (b) Flußdiagramm über den zeitlichen Ablauf der Zustände in einer Arbeitsstation aus [PERF93]

- Bei durchsatzabhängigen Störungen/Pausen ist die Angabe eines Zählbausteins notwendig, dessen Durchsatz für die Auslösung einer Störung bzw. Pause verantwortlich ist. Dieser Zählbaustein muß zu den der Störung bzw. Pause zugeordneten Bausteinen gehören. Die Angabe der Störzeiten bzw. Pausenzeiten erfolgt mit den bekannten Verteilungsfunktionen.

Modellelemente zur Abbildung der Logik:

Bei diesen Strategiebausteinen handelt es sich um die Gesamtheit aller Entscheidungsregeln, Strategien und Ablaufbeschreibungen, die zum Betrieb und für die Lenkung der jeweiligen Fabrik benötigt werden. In PERFECT! gibt es Dispositionsbausteine, Netzwerkbausteine und Prozeßbausteine.

- Die Disposition koordiniert Subsysteme des Fertigungssystems, die darin enthaltenen Netzwerke und die Schnittstellen zu anderen Subsystemen. Die zugehörigen Dispositionsregler verarbeiten Aufträge und Informationen aus der Umgebung des Subsystems und die Zustandsinformationen des Subsystems unter Beachtung von Randbedingungen und Optimierungskriterien zu Anweisungen. Die Anweisungen dienen dem Zweck der Zuweisung von Aufträgen zu Betriebsmitteln und von Betriebsmitteln zu Aufträgen sowie zur Regelung des Verhaltens der Subsysteme. Die Dispositionsebene besitzt die Intelligenz für die flexible Nutzung eines Subsystems im Sinne einer schnellen Anpassung an sich ändernde Umstände. Sie kann auch alternative Ablaufregeln in den Netzwerken anstoßen. Insofern ist hier die Hauptaufgabe zur Regelung anzusiedeln.

Da die Aufgaben der Fabrikregelung auf der Dispositionsebene sehr komplexe Entscheidungen beinhalten, die zumal von menschlichen Entscheidern getroffen werden, wird von PERFECT! die Möglichkeit geboten, diese Regelungsebene mittels der Theorie unscharfer Mengen abzubilden. Mit der Fuzzy-Logic-Control können also Entscheidungen abgebildet werden, die auf vielfältigen (auch gegenläufigen und einander kompensierenden) Kriterien basieren.

- Netzwerke sind zu einer organisatorischen Einheit zusammengefaßte Prozesse. Diese Bereiche beinhalten in der realen Fabrik eine Gruppe von logisch zusammenhängenden Materialfluß- und/oder Fertigungskomponenten. Die zugehörigen Regeleinrichtungen verarbeiten Aufträge und Informationen zu Anweisungen unter Beachtung von Entscheidungsregeln, die es zum Ziel haben, möglichst autonom und ohne Rückkopplung mit übergeordneten Informationsprozessen die Vorgaben der Disposition zu erfüllen. Dazu gehört auch, daß die Netzwerkregeleinrichtungen alternative Ablaufregeln in den Prozessen anstoßen können. In der Netzwerkebene sind also die Vorgaben der Dispositionsebenen möglichst eigenständig umzusetzen und Führungsgrößen an Prozeßsteuerungen zu ermitteln und abzusetzen.

Da auf der Netzwerkebene konkrete Informationen verarbeitet werden, werden hier auch nur scharfe Bedingungen in den Strategien verwendet. Es sind Vorfahrtsstrategien (selbstdefiniert, gemäß Belegungszustand, gemäß Priorität von Objektmerkmalen, gemäß zyklische Vorfahrt) und Verteilstrategien (selbstdefiniert, gemäß Belegungszustand, gemäß automatischer Wegfindung, gemäß Priorität der Ausgänge) für Synchronisationsprozesse implementiert. Zustandsstrategien (selbstdefiniert, Belegungsüberwachung, Merkmalswechsel, Montage/Beladen) können wiederum auf alle Prozeßarten angewandt werden.

• Als Prozesse werden die kleinsten betrachteten Funktionsbereiche bezeichnet. Die zugehörigen SteuerEinrichtungen verarbeiten Aufträge und Informationen zu Anweisungen unter Beachtung vorgegebener Entscheidungsregeln. Diese zielen darauf ab, die Vorgaben des Netzwerkes möglichst autonom und ohne Rückkopplung mit übergeordneten Informationsprozessen zu erfüllen.

Auf Prozeßebene sind Zustandsstrategien (selbstdefiniert, Belegungsüberwachung, Objektgeschwindigkeit, Objektreihenfolge, Merkmalswechsel, Merkmalsänderung, Demontage/Entladung), Verteilstrategien (selbstdefiniert, nach Objektmerkmalen, zyklisch, gewichtet, gemäß Priorität der Ausgänge, gemäß kürzestem Weg) und Vorfahrtsstrategien (selbstdefiniert, gemäß First-Come-First-Served, zyklisch, gewichtet, gemäß Priorität der Eingänge, gemäß kürzestem Weg) angesiedelt.

Das Zusammenspiel der Dispositions-, Netzwerk- und Prozeßstrategien erfolgt in Abhängigkeit von dem augenblicklichen Systemzustand und den Möglichkeiten der Strategien, wobei die Situationen Eindeutigkeitsfall, Mehrdeutigkeitsfall und Konfliktfall unterschieden werden.

Der Fabriksimulator PERFECT! ist auch in der Lage, aus Datenbanken existierende Arbeitspläne zu übernehmen. Das sind sequentielle Auflistungen der Arbeitsgänge, die das jeweilige Werkstück/Produkt zu durchlaufen hat, mit insgesamt jeweils 22 Informationen zu Auftragsgröße, Bearbeitungsort, Fertigungszeit, Ausbeute, Rüstzeit usw.

4.4. SLAMSYSTEM II

<i>Hersteller/Vertreiber:</i>	Pritsker & Associates Corporation, Indianapolis, USA
<i>Preis:</i>	nicht bekannt
<i>Anwendung:</i>	beliebige Simulationsaufgaben
<i>Anwendergruppe:</i>	EDV-Fachleute, Disponenten, mit Simulation vertraute Planer
<i>Hardware-Anforderungen:</i>	i386 oder höher, mind. 2 MB Hauptspeicher, mind. 20 MB freier Festplatten- speicher, VGA-Grafik, OS/2 oder DOS mit Windows

Neutrale Kriterien

Marktpräsenz:

Da SLAMSYSTEM II auf die verbreitete Simulationssprache SLAM II aufgesetzt und von der renommierten Simulationssoftwarefirma Pritsker & Co. entwickelt und vertrieben wurde, besteht ein hoher Marktanteil, insbesondere in den USA.

In Europa wird es zunehmend durch modernere Systeme vom Markt verdrängt. Zum Beispiel durch das Simulationstool FACTOR/AIM aus dem gleichen Softwarehaus.

Ergonomie:

SLAMSYSTEM II bietet eine WINDOWS-Oberfläche mit voller Menü- und Fensterfunktionalität. Die einzelnen Komponenten (Modellerstellung, Parametrierung, Compiler, Linker, Simulator etc.) sind klar getrennt und einzeln aufruf- und austauschbar.

Das System besitzt eine sehr gute On-Line-Hilfe.

Die Kommunikationssprache ist durchweg Englisch. Es gibt ein deutschsprachiges Handbuch.

Erlernbarkeit:

Durch den hohen Abstraktionsgrad erfordert das System von nicht mit Simulationsfragen vertrauten Personen einen hohen Einarbeitungsaufwand. Die strenge Modularisierung gestattet jedoch schnelle Erfolge beim Entwurf und der Simulation kleinerer Modelle, die nicht den vollen Funktionsumfang des Systems nutzen.

Sicherheitsfunktionen:

Es bestehen keine Sicherheitsfunktionen gegen Raubkopieren, unberechtigtes Nutzen, Löschen oder Verändern des gesamten Softwaresystems oder einzelner Komponenten und Modelle.

Informatikspezifische Kriterien

Implementationssprache:

SLAM II auf der Basis der Programmiersprache FORTRAN.
Das System ist offen und modular aufgebaut.

Modellierung:

Ereignis- oder prozeßorientierte Modellsicht mit diskreter, kontinuierlicher oder kombinierter Zeitfortschreibung. Große Modelle können durch End-zu-End-Verknüpfung verbunden werden. Es können keine hierarchischen Modelle erstellt werden. Alle Modelle werden als Dateien im Filesystem des Betriebssystems gespeichert. Modellreplikation und -erweiterbarkeit sind möglich.
Es existiert kein Förderbaustein.

Maximale Modellgröße:

Die Anzahl an Facilities und Entities ist durch das beliebig wiederholbare End-zu-End-Verknüpfen mehrerer Netze theoretisch unbegrenzt. Praktisch limitiert hier nur Speicher- und Rechenzeitbedarf.
Die Attribute sind für jeden Entitytyp festgelegt. Eine Erweiterung ist nicht möglich.

Statistik:

Die Anfangswertbelegung erfolgt über den Modul "Control". Nutzerdefinierte Daten (z.B. Produktionspläne) können nur über die Programmierschnittstelle (FORTRAN oder C) für das System nutzbar gemacht werden.

Zufallszahlengenerierung:

Über den internen Zufallszahlengenerator sind die folgenden Verteilungsfunktionen direkt verfügbar:

- Gleichverteilung $[0,1)$ und $[a,b]$
- Exponentialverteilung
- Dreiecksverteilung $[a,b]$
- Normalverteilung
- logarithmische Normalverteilung
- Erlang-Verteilung
- Gamma-verteilung
- Beta-Verteilung
- Poisson-Verteilung
- Wahrscheinlichkeits-Massenfunktion

Weitere Verteilungen können unter Nutzung des Gleichverteilungszahlenstromes IS mit Hilfe der Programmierschnittstelle hinzugefügt werden.

Robustheit, Verhalten in extremen Situationen:

Das gesamte System ist gegenüber Fehlersituationen, Eingabe- und logischen Fehlern sehr sicher. Die Fehlermeldungen sind genügend präzise und verständlich. Im gesamten Testverlauf trat kein undefinierter Zustand, Programmablauf oder -absturz ein.

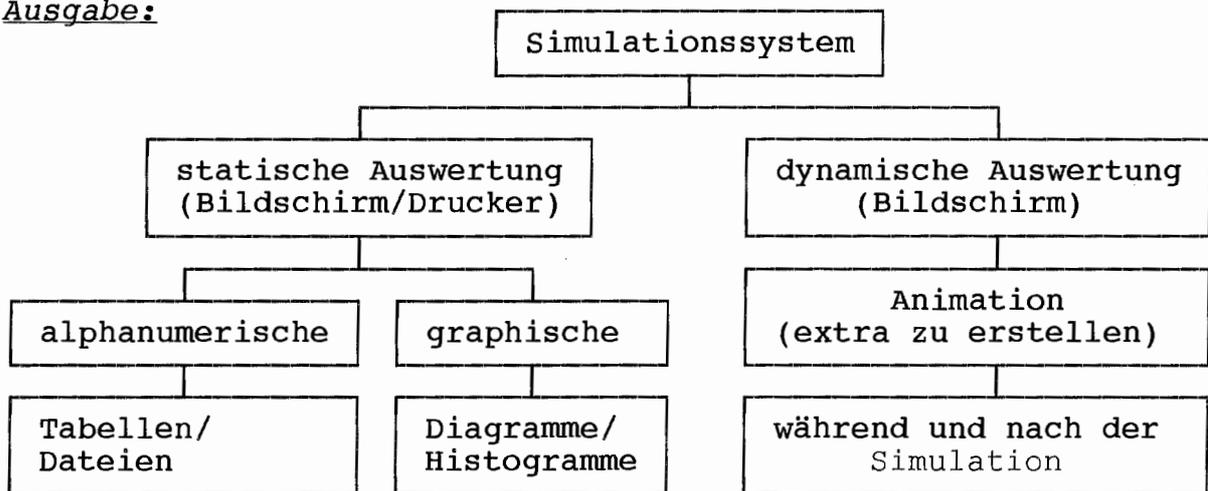
Eingabe:

Der Modellentwurf erfolgt grafisch-interaktiv durch Aufbau und Parametrierung eines Flußdiagramm-Netzwerkes (24 Knotentypen). Ein Entwurf in Textform ist möglich, setzt aber eine gute Syntaxkenntnis voraus. Alphanumerische Eingaben erfolgen über WINDOWS-Eingabemasken mit der Anzeige von Default-Werten.

Die Bedeutung einiger Variablen ist nicht sofort anhand ihres Namens erkennbar und bedarf der Erklärung durch Handbuch oder On-Line-Hilfe.

Die offene Programmierschnittstelle bietet für den versierten Nutzer ein mächtiges Werkzeug zur Erweiterung und Modifikation einzelner Module (Unterprogramme) wie auch des gesamten Systems (neue Module).

Ausgabe:



Die 2D-Animation wird nicht automatisch mit dem Modell erstellt, sondern muß vom Nutzer extra mit Hilfe der Module "Facility" und "Script" entworfen werden. Dieser Mehraufwand hat den Vorteil, daß man nur die für das Simulationsziel relevanten Aspekte einbringen kann.

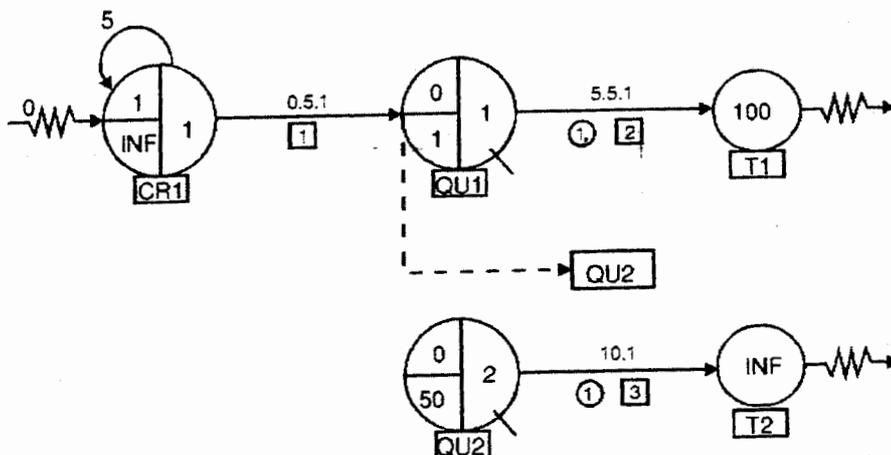


Abb. 4.5 2D-Layout unter SLAMSYSTEM II [SLAM92]

Experimente und Debugging:

Im Netzwerk-Modell kann man auf eine Menge von Variablen und Funktionen zugreifen, mit denen der aktuelle Systemzustand beurteilt und verändert werden kann:

- Zufallszahlenverteilungen
- Statistikfunktionen
- Filemanipulation
- Netzwerkstatusauswertung und -manipulation
- nutzerdefinierte Ausgaben
- Attributmanipulation
- Zugriffsmöglichkeit auf den Systemvektor

Beurteilung des Systems SLAMSYSTEM II in Hinblick der Nutzung für die Zwecke der Heckert-Chemnitzer Werkzeugmaschinen GmbH

Vorteile:

- PC-gestützt, d.h. preiswerte Hardware;
- hohe Flexibilität, d.h. durch entsprechenden Modellier- und Programmieraufwand an jede beliebige Simulationsaufgabe anpaßbar;
- nutzerdefinierte Daten (z.B. Produktionspläne) einsetzbar;
- Schnittstellen zu CAD-Systemen können mit der offenen Programmierschnittstelle realisiert werden;

Nachteile:

- zu hoher Abstraktionsgrad, es existieren keine Entitytypen für Maschinen, Transportsysteme, Lager usw.;
- daraus resultierend ein hoher Einarbeitungsaufwand;
- die 2D-Animation auf diesem Abstraktionsniveau ist für eine ansprechende Präsentation vor Kunden nicht geeignet;

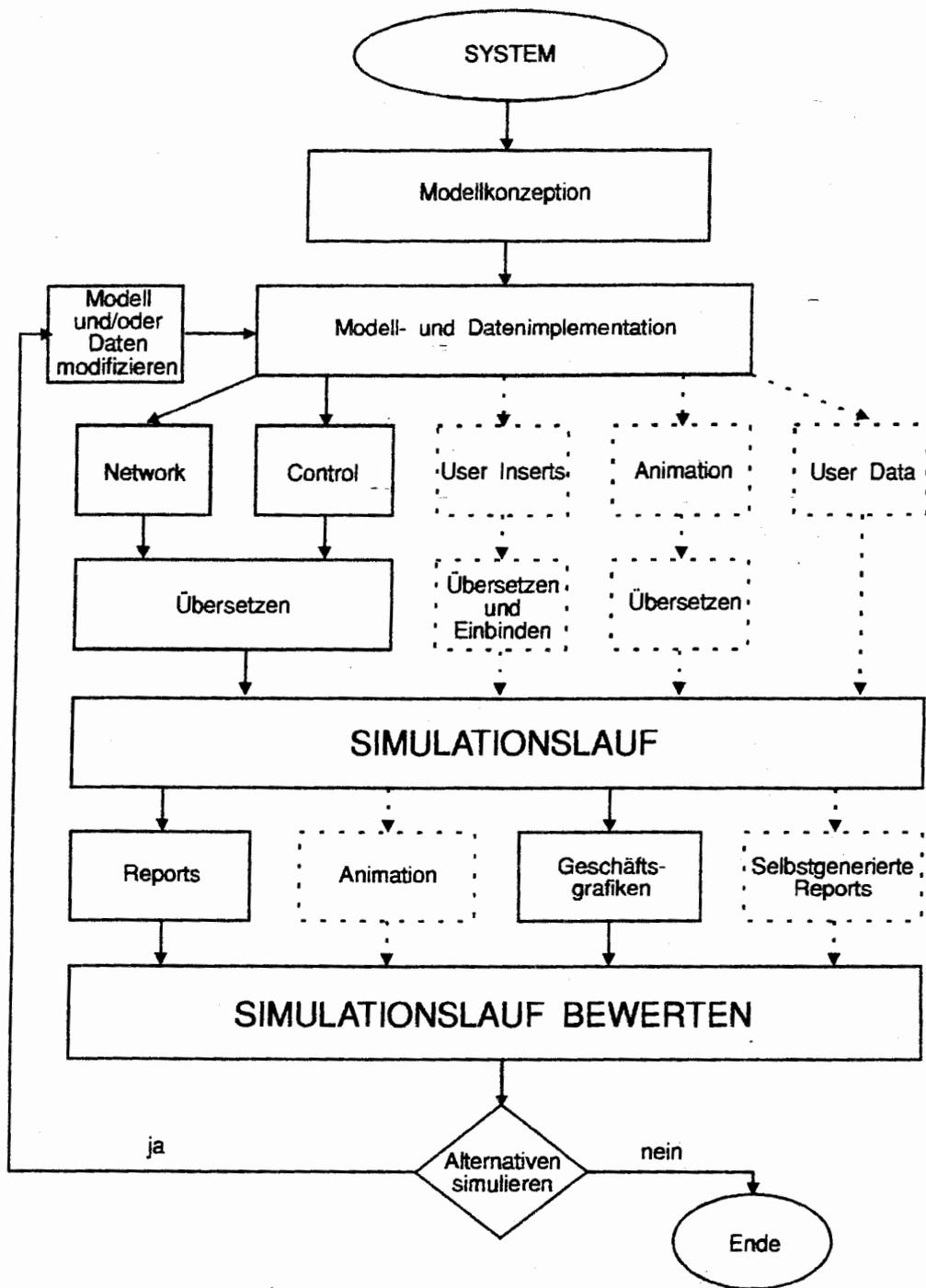


Abb. 4.6 SLAMSYSTEM-Projekttablauf [SLAM92]

Ausführlichere Beschreibung

SLAMSYSTEM ist ein offenes Simulationssystem, das die Abbildung diskreter, kontinuierlicher und kombinierter Prozesse ermöglicht. Zur Abbildung der Funktionalität des zu simulierenden Systems in ein SLAMSYSTEM-Modell werden als zentrale Komponenten ein Netzwerk (Network) und eine Steuerung (Control) benötigt. Weiterhin sind mit SLAMSYSTEM auch eine Generierung von Animationen und die Einbindung von vom Benutzer erstellten Daten und Unterprogrammen möglich.

Das Netzwerk:

Das Netzwerk bildet die Elemente des zu simulierenden Systems mit ihren Eigenschaften und das Zeitverhalten des Systems ab. Es werden logische und zeitliche Verknüpfungen zwischen einzelnen Netzwerksymbolen vorgenommen, so daß ein aus untereinander abhängigen Symbolen bestehendes Systemmodell aufgebaut wird.

- Der CREATE-Knoten wird benutzt, um Entitäten zu generieren.
- Der TERMINATE-Knoten wird benutzt, um Entitäten zu vernichten und/oder die Simulation zu beenden.
- Mit einer REGULAR ACTIVITY wird das Warten von Entitäten bewirkt, um Bedingungen oder Wahrscheinlichkeiten zu prüfen und die Entitäten dann weiterzuleiten.
- Der RESOURCE-Block definiert eine Ressource durch ihren Namen und durch eine Anfangskapazität.
- Der AWAIT-Knoten hält Entitäten zurück, bis eine Ressource frei oder ein Tor offen wird.
- Mit dem FREE-Knoten können belegte Ressourcen verfügbar gemacht werden.
- Der QUEUE-Knoten wird benutzt, um Entitäten in einem File warten zu lassen, bis ein Bearbeiter in der auf den Knoten folgenden Serviceaktivität verfügbar wird.
- Die SERVICE ACTIVITY wird in Verbindung mit dem QUEUE-Knoten benutzt, um eine Warteschlange mit einem Bediener oder eine Warteschlange mit N identischen Bedienern zu modellieren.
- Der COLCT-Knoten wird benutzt, um Statistiken zu erstellen.
- An einem GOON-Knoten wird jede hereinkommende Entität direkt durch den Knoten geschleust.
- Der ASSIGN-Knoten wird benutzt, um SLAM-II-Variablen Werte zuzuweisen.
- Mit einem ALTER-Knoten kann die Kapazität einer Ressource um eine vorgegebene Anzahl von Einheiten verändert werden.

- Der PREEMPT-Knoten wird benutzt, um eine Ressource zu besetzen.
- Der ACCUMULATE-Knoten wird benutzt, um Entitäten zu kombinieren.
- Der BATCH-Knoten wird benutzt, um eine bestimmte Anzahl von Entitäten zusammenzufassen und dann als eine einzige Entität weiterzuleiten. Diese Entität repräsentiert dann das Bündel. Die einzelnen Entitäten des Bündels können mit einem UNBATCH-Knoten wiedergewonnen werden.
- Der UNBATCH-Knoten wird zum Wiedergewinnen einzelner Entitäten, die vorher durch einen BATCH-Knoten in einem Bündel zusammengefaßt waren oder zur Aufsplittung einer Entität in verschiedene, identische Entitäten benutzt.
- Der SELECT-Knoten wird benutzt, um Warteschlangen und verfügbare Bearbeiter auszuwählen.
- Der MATCH-Knoten wird benutzt, um die Bewegung von Entitäten zu verzögern.
- Ein GATE-Block definiert ein Tor durch seinen Namen.
- Mit einem CLOSE-Knoten kann ein Tor geschlossen werden und ein Maximum von M abgehenden Aktivitäten freigegeben werden.
- Der OPEN-Knoten wird benutzt, um ein Tor wieder zu öffnen.
- Der DETECT-Knoten wird benutzt, um immer dann Entitäten zu generieren, wenn eine Variable einen vorgegebenen Schwellenwert in einer bestimmten Richtung überschreitet.
- Mit dem ENTER-Knoten kann der Benutzer in einer Subroutine eine Entität erzeugen und in das Netzwerk übernehmen.
- Der EVENT-Knoten führt dazu, daß bei jeder Ankunft einer Entität eine Subroutine mit einem bestimmten Ereigniscode aufgerufen wird. Dieses erlaubt dem Benutzer, Funktionen zu modellieren, für die kein Standardknoten zur Verfügung steht. Ein Maximum von M abgehenden Aktivitäten wird initiiert.

Jeder Knoten muß durch Eingabe einer Anzahl gewisser spezifischer Parameter genau in seiner Funktionalität beschrieben werden. Dazu sind auch bestimmte Standardwerte vorgegeben.

Im Netzwerk kann auf eine Vielzahl von Variablen und Funktionen zugegriffen werden, mit denen der aktuelle Systemzustand beurteilt und verändert werden kann. Damit wird auch die Abbildung sehr komplexer Logiken möglich.

Weiterhin gibt es einen globalen Systemvektor und eine globale Systemmatrix, in denen Daten gespeichert werden können, die damit im gesamten Modell sichtbar, zugreifbar und manipulierbar sind.

Bei der Modellerstellung kann auch mit Zufallsvariablen gearbeitet werden, die jeweils Pseudozufallszahlen gemäß einer bestimmten statistischen Verteilung beinhalten.

Die Steuerung:

Zur Initialisierung und Steuerung von Simulationsläufen werden Simulationsparameter gesetzt (z.B. Dauer eines Simulationslaufes), Initialbedingungen definiert (z.B. Zustand von Systemkomponenten im Netzwerk) und Festlegungen über die von der Simulation zu generierenden Ausgaben und Auswertungen getroffen. Dafür stehen eine Reihe von Kommandos zur Verfügung.

- Das ARRAY-Kommando wird benutzt, um eine Reihe der Systemmatrix zu definieren.
- Das CONTINUOUS-Kommando spezifiziert, daß kontinuierliche Variablen evaluiert werden, und definiert die Größe der Zeitschritte, in denen die kontinuierlichen Variablen aktualisiert werden.
- Das ENTRY-Kommando wird benutzt, um zu Simulationsbeginn Einträge in Files zu initiieren.
- Das EOUVALENCE-Kommando wird benutzt, um SLAM-II-Variablen und Zufallsvariablen Namen zuzuordnen. Damit soll die Lesbarkeit und Veränderbarkeit des Modells erleichtert werden.
- Das FIN-Kommando definiert das Ende aller SLAM-II-Eingabekommandos. Alle nachfolgenden Läufe werden ohne weitere Dateneingabe in SLAM II durchgeführt.
- Das GEN-Kommando ist immer das erste Kommando. Es gibt allgemeine Informationen, die das Modell, die Anzahl von Abläufen und das Ausgabeformat identifizieren.
- Das INITIALIZE-Kommando wird benutzt, um die Anfangs- und Endzeit für eine Simulation zu spezifizieren und die Initialisierungsoptionen zu setzen für Variablen, Files und Statistiken.
- Das INTLC-Kommando wird benutzt, um verschiedenen globalen Variablen Initialwerte zuzuordnen. Der Initialwert muß eine Konstante sein. Mit einem Kommando können Mehrfachzuweisungen vorgenommen werden.
- Das LIMITS-Kommando wird benutzt, um Größe und Struktur des Filesystems zu definieren. Es folgt dem GEN-Kommando.
- Das MONTR-Kommando wird benutzt, um ausgewählte Zwischenergebnisse zu drucken, den interaktiven Ablauf zu initiieren oder die Statistikwerte nach einer Anlaufzeit zu löschen.
- Das NETWORK-Kommando wird benutzt, um den Anfang einer Netzwerkbeschreibung anzuzeigen. Dieses Kommando kann auch zum Sichern oder Nachladen eines übersetzten Netzwerkes benutzt werden.

- Das PRIORITY-Kommando wird benutzt, um die Reihenfolgeregel für ein File zu spezifizieren. Wenn diese Regel nicht durch ein PRIORITY-Kommando ersetzt wird, ist sie FIFO für alle Files (außer dem Ereignisfile). Die Reihenfolgeregel des Ereignisfiles basiert auf der Ereigniszeit und besagt, daß der kleinste Ereigniszeitpunkt Priorität hat. Diese Regel kann nicht verändert werden. Im Falle gleicher Ereigniszeitpunkte tritt die FIFO-Regel als zweites Entscheidungskriterium im Kraft. Diese Regel ist veränderbar.

- Das RECORD-Kommando wird benutzt, um einen Plot (Datenaufzeichnung) anzufordern. Es definiert die unabhängige Variable, das Speichermedium und die Plotzeitpunkte. Dem RECORD-Kommando folgt ein Satz von VAR-Kommandos, die die abhängigen Variablen des Plots definieren.

- Das SEEDS-Kommando wird benutzt, um den Startwert eines Zufallszahlenstromes zu ersetzen und damit Zufallszahlenströme für verschiedene Läufe zu kontrollieren.

- Das SEVNT-Kommando wird benutzt, um die Kontrolle an eine vom Benutzer geschriebene Ereignissubroutine zu übergeben. Dies geschieht immer dann, wenn eine spezifizierte Variable einen vordefinierten Wert in einer bestimmten Richtung überschreitet. Dabei kann auch eine Toleranz mit berücksichtigt werden. Dann wird die Subroutine nur ausgelöst, wenn die beobachtete Änderung des Wertes der Variablen innerhalb der Toleranz liegt. Mit dem DETECT-Knoten erlaubt SLAM II, daß die Variable sowohl diskret als auch kontinuierlich sein darf.

- Das SIMULATE-Kommando wird benutzt, wenn mehrfach Simulationsläufe durchgeführt werden, zwischen denen Daten gelesen werden müssen. Ein Simulationslauf wird für die Kommandos durchgeführt, die vor dem SIMULATE-Kommando stehen. Diesem kann jede Art von Datenaktualisierung folgen (z. B. Neuinitialisierung des Zufallszahlenstromes, ENTRY-Daten für weitere Simulationsläufe). Danach wird ein weiterer Simulationslauf gestartet.

- Das STAT-Kommando wird zur Definition einer Variablen benutzt, für die durch Aufruf einer vom Benutzer geschriebenen Subroutine Statistiken gesammelt werden sollen.

- Das TIMST-Kommando wird benutzt, um die automatische Sammlung von zeitabhängigen Statistiken über die globalen Variablen oder die Statusvariablen bzw. abgeleiteten Variablen auszulösen. Es können Histogramme angefordert werden, die die prozentualen Zeitanteile, in denen sich die Werte der Variablen in bestimmten Intervallen befanden, auswerten.

Jedes Kommando erfordert die Eingabe einer Anzahl gewisser spezifischer Parameter.

Unterprogramme in SLAMSYSTEM:

SLAMSYSTEM bietet dem Benutzer die Möglichkeit, Unterprogramme und Funktionen in den Programmiersprachen FORTRAN oder C selbst zu entwickeln und in das Simulationsmodell einzubinden. Im Netzwerk kann dann auf diese sowie eine ganze Anzahl bereits vorhandener Programme und Funktionen zugegriffen werden.

Dem Nutzer werden von SLAMSYSTEM bereits Programme und Funktionen zur Verfügung gestellt (für Zufallsverteilungen, Statistikfunktionen, Filemanipulationen, Netzwerkstatusauswertungen und -manipulationen, zur Unterstützung nutzereigener Ausgaben). Selbst erstellte Routinen können durch von SLAMSYSTEM bereitgestellte Schnittstellen in das Simulationsmodell eingebunden werden. Diese Programme und Funktionen sind unter einem definierten Namen mit festgelegter Syntax zu erstellen bzw. aufzurufen. Beim Übersetzen eines Simulationsmodells werden Dummy-Versionen aller dieser Programme eingebunden, die bei Vorhandensein einer vom Nutzer geschriebenen Routine automatisch durch diese ersetzt werden.

4.5. MOSYS

<i>Hersteller:</i>	FHG-IPK Fraunhofer-Institut für Produktionsanlagen und Konstruktionstechnik, Bereich Planungstechnik Pascalstraße 8-9, 10587 Berlin
<i>Vertreiber:</i>	siehe Hersteller
<i>Preis:</i>	20-50 TDM
<i>Anwendung:</i>	Produktion, Logistik, Materialfluß
<i>Anwendergruppe:</i>	mit Simulation vertraute Planer
<i>Hardware-Anforderungen:</i>	Mainframe (IBM-VM mit CMS), diverse Workstations

Neutrale Kriterien

Marktpräsenz:

Da MOSYS das bei weitem älteste der hier untersuchten Systeme ist, kann auf eine relativ große Verbreitung geschlußfolgert werden. Allerdings ist es in seinen Basiskonzepten veraltet und wird deshalb zunehmend durch modernere Systeme vom Markt verdrängt.

Ergonomie:

MOSYS besitzt eine grafisch-interaktive Benutzeroberfläche mit voller Fenster- und Mausfunktionalität. Leider besteht keine durchgehende Nutzerführung, sondern während der Modellbearbeitung ist ein zeitaufwendiges Wechseln zwischen den einzelnen Modulen erforderlich (längere Speicher- und Ladezeiten), was sich besonders beim Auftreten von Fehlern störend bemerkbar macht.

Es existiert eine gute On-Line-Hilfe. Die Kommunikationssprache ist durchweg Deutsch. Es gibt ein deutschsprachiges Handbuch.

Erlernbarkeit:

Zwar existieren mit den 5 Funktionsbausteinen "Fertigen", "Fördern", "Lagern", "Prüfen" und "Montieren" relativ realitätsnahe Modellierobjekte, aber deren zu abstrakte Darstellung erschwert die Layoutgestaltung erheblich. Obwohl für die Nutzung keine informatikspezifischen Kenntnisse notwendig sind, ist der Einarbeitungsaufwand äußerst hoch: Die Dokumentation umfaßt elf Bände!

Sicherheitsfunktionen:

Es bestehen keine Sicherheitsfunktionen gegen Raubkopieren, unberechtigtes Nutzen, Löschen oder Verändern des gesamten Softwaresystems oder einzelner Komponenten und Modelle. Lediglich in UNIX und VMS können die üblichen Zugriffsrechte und Schutzdienste des Betriebssystems genutzt werden.

Eine automatische Sicherungsfunktion bei Systemabsturz oder vor Simulationsläufen ist nicht implementiert, so daß der Nutzer vor den Simulationsläufen explizit Sicherungskopien anfertigen sollte.

Informatikspezifische Kriterien

Implementationssprache:

Über die Implementationssprache liegen uns keine Informationen vor. Das System ist geschlossen und modular aufgebaut.

Modellierung:

Ereignisorientierte Modellsicht mit diskreter Zeitfortschreibung. Dabei setzt sich ein Modell aus Funktionsbausteinen, Steuerstrategien und bewegten Objekten zusammen.

Den einzelnen Funktionsbausteinen werden im Editier-Modus Parameter in Form von Attributen übergeben. Dazu gehören z.B. Name, Kapazität, Rüstzeit oder Störverhalten.

Die Steuerstrategien werden über Token eingegeben. Dazu wird in einer Maske eine Liste möglicher Token mit Eigenschaften definiert, die dann in einer Attributmaske den jeweiligen Elementen zugeordnet werden. Durch gezielte Manipulation zu bestimmten Zeitpunkten ist so auch eine globale Steuerstrategie möglich. Bewegte Objekte werden in Objektlisten mit Quelle, Senke und Art der Einlastung vereinbart.

Alle Modelle werden in einer Modelldatenbank gespeichert. Modellreplikation und -erweiterbarkeit sind möglich. Es können hierarchischen Modelle erstellt werden.

Maximale Modellgröße:

Durch das Bausteinkonzept ist die Modellgröße theoretisch unbegrenzt. Praktisch limitiert hier nur Speicher- und Rechenzeitbedarf.

Statistik:

Für die statistische Auswertung stehen die Module AUSWERTE und SIMSTAT zur Verfügung. Die mit diesen Modulen erstellten Tabellen und Dateien können mit der Tabellenkalkulation EXCEL weiterverarbeitet werden.

Das Generieren selbstdefinierter Statistiken ist nicht möglich. Die Einschwingphase muß der Nutzer nach eigenen Kriterien ermitteln. Er kann diese dann mit der Definition von Auswertungszeiträumen aus der Statistik ausblenden.

Zufallszahlengenerierung:

Der Zufallszahlengenerator liefert Verteilungsfunktionen für Störungen (negative Exponentialverteilung), Nacharbeit und zufällige Einlastung, wobei diese Funktionen und deren Werte nicht näher beschrieben sind.

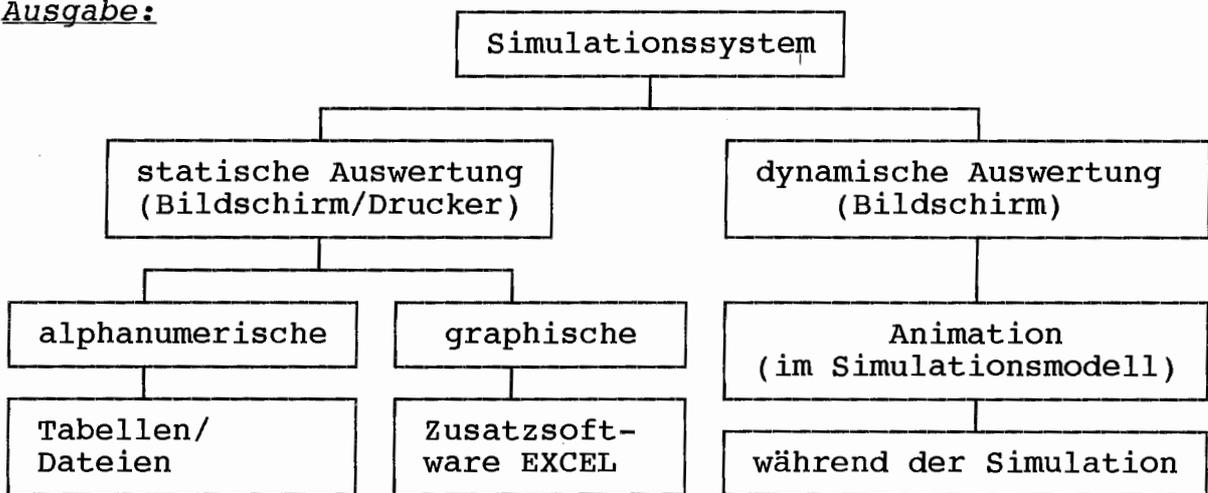
Robustheit, Verhalten in extremen Situationen:

Das gesamte System ist gegenüber Fehlersituationen, Eingabe- und logischen Fehlern sehr sicher. Die Fehlermeldungen sind genügend präzise und verständlich. Im gesamten Testverlauf trat kein undefinierter Zustand, Programmablauf oder -absturz ein.

Eingabe:

Der Modellentwurf erfolgt grafisch-interaktiv durch Aufbau und Parametrierung eines Flußdiagramm-Netzwerkes (5 Bausteine). Dabei können beliebige Modellausschnitte vergrößert werden. Das Setzen der Parameter und Steuerstrategien geschieht durch das Ausfüllen von Parametermasken. Es ist ein Editieren, Kopieren und Löschen von Elementen und Arbeitsplänen möglich. Zusätzlich verfügt MOSYS noch über eine eigene Kommandosprache mit der dem Anwender eine Programmierschnittstelle zur konkreten Anpassung des Systems an eine Simulationsaufgabe angeboten wird.

Ausgabe:



Die 2D-Animation wird automatisch mit dem Modell erstellt und kann durch den Nutzer für einzelne Elemente oder das ganze Modell definiert werden. Mit ihrer Hilfe kann das Modell verifiziert und das Entstehen kritischer Situationen (z.B. Deadlock) analysiert werden. Die Animation kann zur Beschleunigung der Simulation abgeschaltet werden. Eine 3D-Animation ist über eine CAD-Schnittstelle möglich.

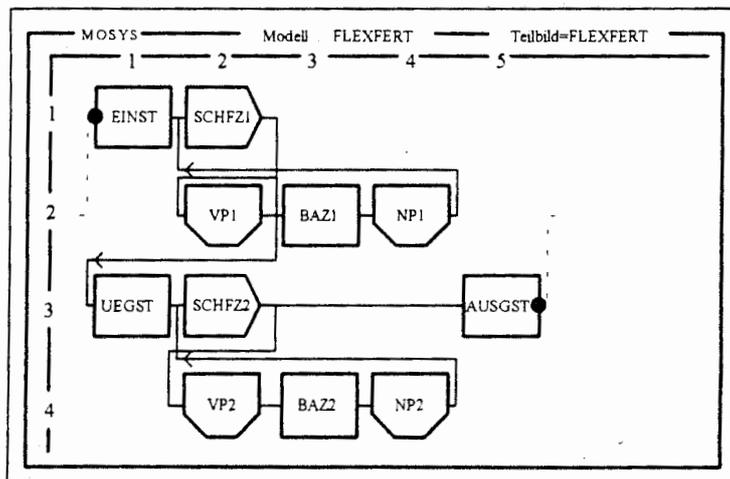


Abb. 4.7 2D-Layout unter MOSYS [MOSY93]

Experimente und Debugging:

Über den Menüpunkt "Prüfen" des Moduls TRAMEIN ist ein Konsistenzcheck möglich. Durch die interaktive Simulation kann das Modell über einen auswählbaren Zeitraum detailliert betrachtet werden. Die Abfrage aktueller Zustandsinformationen erfolgt über einblendbare Fenster. Während der Simulation sind die Betriebsmodi:

- Batchbetrieb (alle Anzeigen ausgeschaltet)
- Dialogbetrieb (nur Animation ausgeschaltet)
- Animationsbetrieb (umfangreiche Eingriffsmöglichkeiten)

wählbar.

Für den erfahrenen Anwender bietet die MOSYS-Kommandosprache vielfältige Möglichkeiten zur Fehlersuche. So können Funktions- und Systemzustände abgefragt und Plausibilitätskontrollen durchgeführt werden.

Beurteilung des Systems MOSYS in Hinblick der Nutzung für die Zwecke der Heckert-Chemnitzer Werkzeugmaschinen GmbH

Vorteile:

- niedriger Abstraktionsgrad, es existieren Bausteine aus der realen Welt des Materialflusses;
- nutzerdefinierte Daten (z.B. Arbeitspläne) einsetzbar;
- gute 2D-Animation mit vielen Eingriffsmöglichkeiten;
- die Schnittstelle zu CAD-Systemen erlaubt eine 3D-Animation;

Nachteile:

- großes System mit finanziell hohem Aufwand für Soft- und Hardware;
- in seinen Basiskonzepten veraltet und zu langsam bei der Modellerstellung;
- hoher Einarbeitungsaufwand;
- geringe Flexibilität, da der sehr begrenzte Bausteinumfang nicht alle Elemente moderner Fertigungssysteme abbildet;
- die 2D-Animation mit sehr abstrakten Symbolen ist für eine ansprechende Präsentation vor Kunden nicht geeignet;

4.6. TAYLOR II

<i>Hersteller:</i>	F&H Logistics and Automation B.V. Sporlaan 424, 5038 CG Tilburg (NL)
<i>Vertreiber:</i>	GDC Gesellschaft für DV-Beratung & Entwicklung Am Kaunert 2g, 41189 Mönchengladbach
<i>Preis:</i>	10-20 TDM
<i>Anwendung:</i>	Produktion, Logistik, Materialfluß
<i>Anwendergruppe:</i>	mit Simulation vertraute Planer und Disponenten
<i>Hardware-Anforderungen:</i>	PC i486 oder Pentium mit MS-DOS ab Version 5.0, mind. 4 MB Hauptspeicher, mind. 5 MB freier Festplatten- speicher, VGA-Grafik mit IBM8514-Treiber, Maus

Neutrale Kriterien

Marktpräsenz:

Das in den Niederlanden entwickelte System hat eine breite Akzeptanz in kleinen und mittelständischen Firmen, sowie an Universitäten und Fachhochschulen gefunden. Sowohl Preis als auch Hardwareanforderungen ordnen es in die mittlere Klasse der Produktionsplanungs- und Simulationssysteme ein. Sein Einsatz für kleine und mittlere Simulationsaufgaben vermittelt einen guten Eindruck von den Konzepten und dem Leistungsvermögen moderner Simulationssoftware.

Ergonomie:

TAYLOR II besitzt eine grafisch-interaktive Benutzeroberfläche mit voller Fenster- und Mausfunktionalität. Damit wird der Nutzer durch die Mensch-Maschine-Kommunikation zielgerichtet zu einem gültigen und lauffähigen Simulationsmodell geführt. Die Benutzeroberfläche orientiert sich leider nicht an einem der heute für PC's gebräuchlichen Standards (SAA, ACE, Windows). Eine nur optisch vorhandene Ähnlichkeit mit GEM schließt nicht aus, daß besonders in der Wahl von Farben, Schriftarten und Linienstärken noch bessere Optionen getroffen werden könnten. Auf manchen VGA-Bildschirmen waren die Konturen zu schwach, um länger als eine Stunde ohne Beeinträchtigung der optischen Wahrnehmung arbeiten zu können.

Negativ fällt auch die "anglo-deutsche" Kommunikationssprache auf. So sind Funktionen in Menüs teilweise Englisch, teilweise Deutsch benannt (z.B. "Startwarm", "Autofit", "P&A"). Die meisten Dialoge laufen jedoch in Deutsch ab. So auch die On-Line-Hilfe. Lediglich Fehlermeldungen erfolgen durchweg in Englisch. Es gibt ein deutschsprachiges Handbuch.

Erlernbarkeit:

Durch den geringen Abstraktionsgrad ermöglichen die Grundbausteine (Maschine, Transporter, Puffer, Hilfe, Förderer, Weg, Lager, Reservoir, Ein-/Ausgabe) eine rationale Layoutgestaltung, da sie einerseits der realen Materialflußwelt nachempfunden wurden und andererseits über eine menügeführte Benutzeroberfläche positioniert und parametrisiert werden. Für die Nutzung sind keine informatikspezifischen Kenntnisse notwendig. Die Dokumentation enthält eine Reihe Lernbeispiele, die zwar den Einstieg erleichtern, aber für spezielle Probleme nicht immer ausreichend sind. Die On-Line-Hilfe mit thematischer Suchfunktion ist eine gute Ergänzung zum Handbuch. Wir schätzen den Einarbeitungsaufwand für mit Simulation vertraute Personen gering, für Anfänger auf diesem Gebiet mittelmäßig ein.

Sicherheitsfunktionen:

Das gesamte System ist mit einem Hardware-Dongle am Parallelport gegen Raubkopieren geschützt. Ein zusätzlicher Paßwortschutz gegen unberechtigtes Nutzen, Löschen oder Verändern des gesamten Softwaresystems oder einzelner Komponenten und Modelle besteht nicht.

Als ausgesprochen positiv sind die Sicherheitsfunktionen gegen Datenverlust hervorzuheben. So wird bei einem Systemabsturz das aktuelle Simulationsmodell in der Datei RESCUE.SIM gespeichert. Zusätzlich kann eine intervallweise Speicherung während der Bearbeitungsphase aktiviert werden. Hier kommen eindeutig die Konzepte moderner Programmierungstechniken (DOS-Protected-Mode) zum Tragen.

Informatikspezifische Kriterien

Implementationssprache:

TAYLOR II ist in der Programmiersprache BORLAND PASCAL 7.0 geschrieben worden. Es läuft unter DOS mit dem DOS-Protected-Mode-Interface (DPMI).

Das System ist offen, aber nicht modular aufgebaut.

Modellierung:

Ereignisorientierte Modellsicht mit diskreter Zeitfortschreibung.

Die Modellwelt des Simulators gliedert sich in 5 Ebenen:

- Elemente (9 Bausteintypen)
- Jobs (Arbeitsgänge)
- Routen (Wege im Layout)
- Schritte (Bearbeitungs- oder Transportvorgang)
- Produkte (Werkstücke)

Lokale Steuerstrategien werden über den Menüpunkt "Parameter-Route" eingegeben. Die Abbildung globaler Steuerstrategien erfolgt über die Manipulation von Attributen mit Hilfe der "Trigger-Funktion". Während der Simulation sind die Produktcodes ebenfalls mittels der Trigger-Funktion änderbar, so daß der Produktwandel (Fertigungsfortschritt) grafisch darstellbar wird.

Alle Modelle werden im Filesystem gespeichert. Modellreplikation und -erweiterbarkeit sind möglich. Es können keine hierarchischen Modelle erstellt werden.

Maximale Modellgröße:

Durch das Bausteinkonzept ist die Modellgröße theoretisch unbegrenzt. Praktisch limitiert hier nur Speicher- und Rechenzeitbedarf. Dazu werden im Handbuch umfangreiche Überlegungen angestellt. So sollte die maximale Systemgröße von 1200 Attributen in einem Modell nicht überschritten werden. Insbesondere wird bei großen Modellen die Einrichtung einer RAM-Disk empfohlen.

Statistik:

Über das gesamte Modell und jedes Modellelement kann eine genaue Statistik geführt werden. Dabei sind besonders die nahezu unbegrenzten Möglichkeiten einer benutzerdefinierten Statistik sowohl während als auch nach der Simulation hervorzuheben. Über die Funktion "Parameter-Vorrat" kann das Modell initialisiert werden.

Eine Einmaligkeit unter den untersuchten Systemen steht in TAYLOR II mit den Funktionen:

- Modell analysieren (Konfidenz- und Korrelationstests)
- Einschwingtest (ein Welsh-Graph zeigt die Länge der Einschwingphase an)
- Optimieren, Sensitivitätsanalyse

zur Verfügung.

Deren Wirksamkeit und mathematische Exaktheit sollte noch Gegenstand weiterer Untersuchungen sein.

Nutzerdefinierte Daten (z.B. Arbeitspläne, Fertigungsprogramme) können über die Programmierschnittstelle (in PASCAL) unter Nutzung des TAYLOR-Language-Interface (TLI) für das System nutzbar gemacht werden.

Mit den separaten Modulen "Advanced Statistics" und "Unifit" können zwei der besten und modernsten Statistik-Softwarepakete die Daten für und von TAYLOR II aufbereiten bzw. analysieren. Diese Produkte müssen aber zusätzlich erworben werden.

Zufallszahlengenerierung:

Aus der mit dem Zufallszahlengenerator von BORLAND PASCAL erzeugten Gleichverteilung werden 20 verschiedene Verteilungsfunktionen generiert. Sollten diese noch nicht ausreichen, kann mit Hilfe von TLI jede beliebige Verteilung implementiert werden. Zusätzlich unterstützt TAYLOR II mit dem Befehl "Verteil" empirische Verteilungen in Form von Häufigkeitstabellen und Textdateien.

Robustheit, Verhalten in extremen Situationen:

Das gesamte System ist bei korrekter Installation und ausreichender Speicherkonfiguration gegenüber Fehlersituationen, Eingabe- und logischen Fehlern relativ sicher. Die englischen Fehlermeldungen könnten allerdings ausführlicher und präziser sein. Unter Umständen treten noch Fehlermeldungen des BORLAND PASCAL-Laufzeitsystems auf, was auf mangelhafte Programmierungstechnik hindeutet. So wird bei der Arbeit mit Routing-Dateien ohne vorherige Verbindung der Routen-Elemente der DPML-Fehler 216 "Allgemeine Schutzverletzung" durch Dereferenzierung von nil-Zeigern gemeldet - Folge: Systemabsturz.

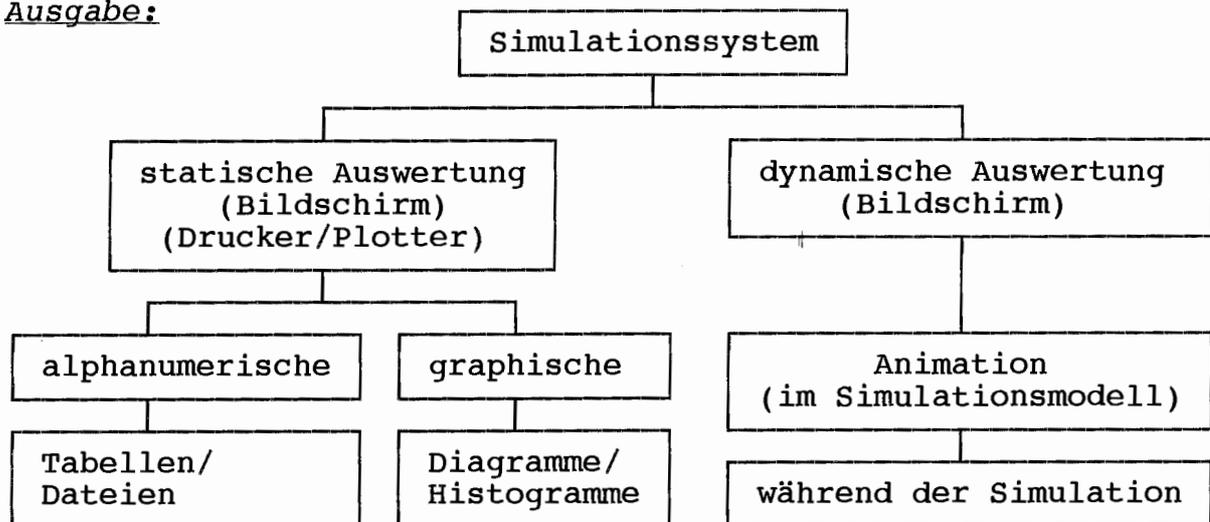
Eingabe:

Der Modellentwurf erfolgt grafisch-interaktiv durch Aufbau und Parametrierung eines Flußdiagramm-Netzwerkes (9 Bausteine). Dabei können beliebige Modellausschnitte vergrößert werden. Das Setzen der Parameter geschieht durch das Ausfüllen von Parametermasken. Die einzelnen Bausteine können auch durch eine Routingdatei, separat erzeugt in einem Editor, logisch miteinander verknüpft werden.

Der Änderungsdienst ist umständlich und gewöhnungsbedürftig. Das Einfügen und Löschen von Elementen gestaltet sich kompliziert. Das Kopieren von bereits parametrierten Elementen ist nicht möglich, jedoch können Elementeparameter teilweise oder vollständig auf andere Elemente übertragen werden.

Im wesentlichen ist eine flexible Modellgestaltung ohne Nutzung der Programmierschnittstelle TLI nicht möglich. TLI ist, als eine Sammlung von PASCAL-Routinen, für Anwender mit Kenntnissen dieser Programmiersprache kein Problem.

Ausgabe:



Die 2D/3D-Animation wird automatisch mit dem Modell erstellt. Dabei hat der Nutzer weitreichende Definitionsmöglichkeiten, bis hin zur Kreation eigener Icons für die Elemente (Maschinen, Transporter, Werkstücke etc.). Insbesondere die dynamische Icondarstellung zur Abbildung des Produktwandels während des Fertigungsprozesses ist hervorzuheben. Die graphischen Auswertemöglichkeiten sind nahezu unbegrenzt und lassen kaum Wünsche offen. Sollte diese dennoch auftreten, so kann über die TLI-Programmierschnittstelle und nachgeschaltete Auswertesoftware jede benutzerdefinierte Ausgabe realisiert werden.

Experimente und Debugging:

Neben der sehr aussagekräftigen On-Line-Animation, kann das Simulationsmodell im Trace-Modus schrittweise abgearbeitet und alle Variablenwerte direkt abgefragt werden.

Ein Konsistenzcheck (z.B. für die Routing-Datei) erfolgt nicht. Über den Befehl "Simuliere-Batchrun" werden mehrere Experimente ohne Zwischenstop hintereinander ausgeführt. Der Modul "Dokument" bietet Werkzeuge zur Auswertung von Simulationsdateien (Snapshots) an. Insgesamt gestaltet sich der Ablauf einer Simulation für den Nutzer sehr transparent, womit die Fehlersuche sehr erleichtert wird.

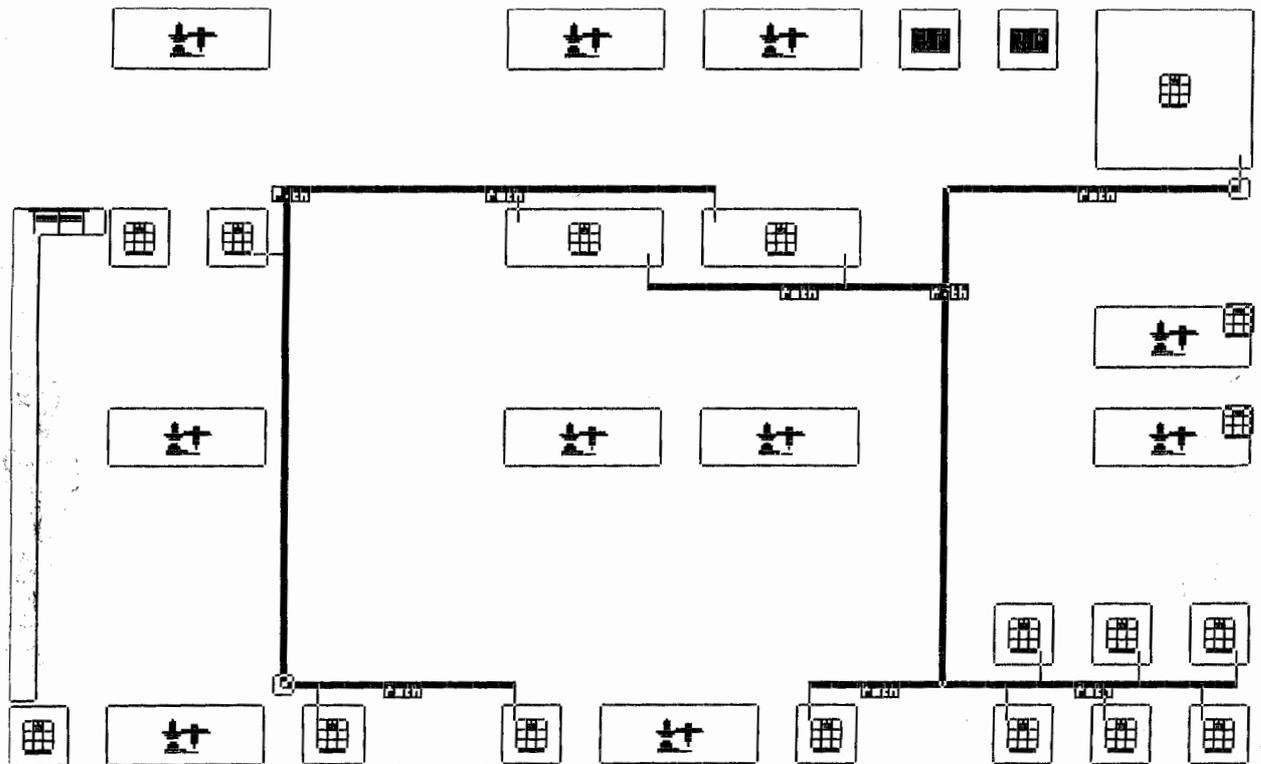


Abb. 4.8 TAYLOR II-Layout der Fabrik "Getriebefertigung."

Beurteilung des Systems TAYLOR II in Hinblick der Nutzung für die Zwecke der Heckert-Chemnitzer Werkzeugmaschinen GmbH

Vorteile:

- preiswerte Hard- und Software;
- niedriger Abstraktionsgrad, es existieren Bausteine aus der realen Welt des Materialflusses;
- daraus resultierend niedriger Einarbeitungsaufwand;
- hohe Flexibilität, d.h. durch entsprechenden Modellieraufwand an jede Materialflusssimulation anpaßbar;
- nutzerdefinierte Daten (z.B. Arbeitspläne) einsetzbar;
- gute, aussagekräftige 2D/3D-Animation mit selbstgestaltbaren Icons;
- dynamische Icondarstellung (Produktwandel);
- sehr umfangreiche grafische Auswertemöglichkeiten;
- das Angebot eines "Runtime-Moduls" ermöglicht dem Nutzer, sich Modelle als Dienstleistung erstellen zu lassen, ohne selbst das komplette Simulationssystem erwerben zu müssen;

Nachteile:

- Ergonomie und Bedienbarkeit entsprechen nicht den Anforderungen moderner Softwaresysteme;
- die Software enthält noch grobe Programmierfehler;
- sind größere Systeme zu simulieren, so macht sich die relativ langsame Simulationsgeschwindigkeit störend bemerkbar (Pentium-Prozessor empfehlenswert);
- die Animation kann nicht als "Film" gespeichert werden, so daß eine Simulation vor Kunden sehr zeitaufwendig ist.

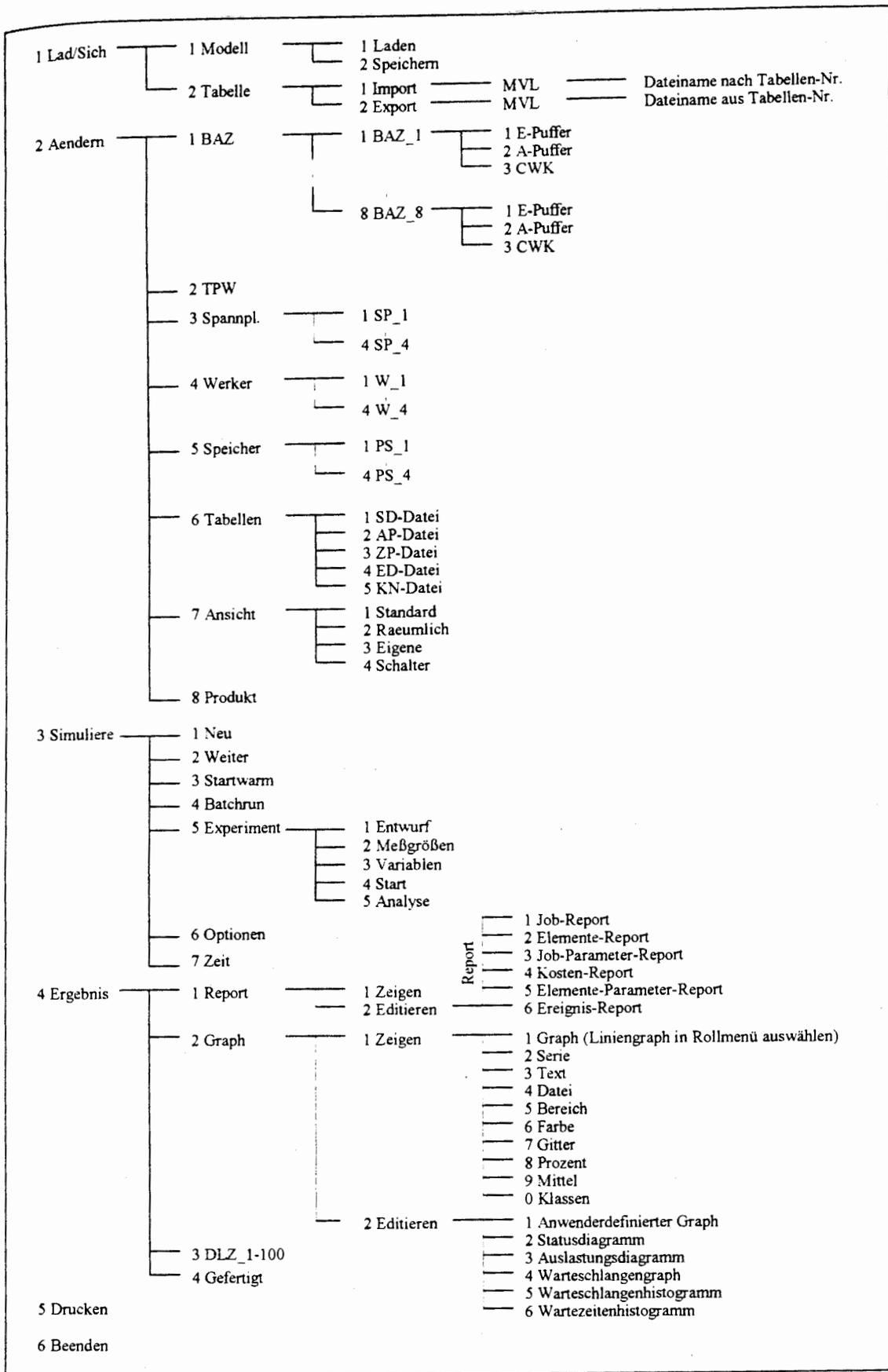


Abb. 4.9 Menüverzweigung im Runtime-Modul [ZIMM94]

Ausführlichere Beschreibung

Mit TAYLOR II können reale Systeme, hauptsächlich aus Produktion und Logistik, im Computer modelliert und simuliert werden. Dazu gibt es die fünf grundlegenden Einheiten Elemente, Jobs, Routen, Schritte und Produkte zum Aufbau eines Modells. Zum Detaillieren des Modells sind Parameter anzugeben, die die betreffenden Einheiten und damit das Modell entsprechend dem realen System genau beschreiben. Hat das reale System jedoch ein spezielles Verhalten, das nicht mit den Einheiten und Parametern von TAYLOR II beschreibbar ist, so bietet TAYLOR II mit der Schnittstelle TLI fast uneingeschränkte Möglichkeiten, damit die Dinge im Modell auch so funktionieren, wie sie im realen System ablaufen.

Elemente:

Die Modellerstellung beginnt bei der Definition des Layouts, d.h. die das Modell bildenden Elemente werden als Icons auf dem Bildschirm plazierte.

Bei der abstrakten Beschreibung von Produktionsprozessen gilt es, die Aspekte Lagerung, Bearbeitung und Transport darzustellen.

In TAYLOR II gibt es fünf Elementtypen, die die Lagerung darstellen.

- Der Puffer ist die allgemeine Darstellung für die Lagerung.
- Das EinAus-Element hat eine Quelle-Senke-Funktion und wird üblicherweise am Beginn oder am Ende einer Produktionslinie eingesetzt. Es generiert oder entfernt Produkte.
- Obwohl TAYLOR II ein Simulator für diskrete Ereignisse ist, kann man Prozesse simulieren, die einige Aspekte eines kontinuierlichen Systems aufweisen, indem man einen kontinuierlichen Fluß in kleine Einheiten aufteilt. Die Standardeinstellung des Reservoirs zeigt anstelle von Produkten Füllstände und besitzt die Möglichkeit, Schaltmarken (Level) zum Öffnen und Schließen des Reservoirs anzugeben.
- In einem Lager können die Produkte an einen bestimmten Lagerplatz (Zeile, Spalte) gesetzt werden.
- Der Förderer ist eine Kombination von Lager- und Transport-Element. Er wird eingesetzt, wenn eine Verzögerung am Puffer notwendig ist, bevor das Produkt den Puffer verläßt, und ist somit eine Form des kontinuierlichen Transports.

Es gibt drei Elementtypen, um die Bearbeitung darzustellen.

- Die Maschine ist die allgemeine Darstellung für eine Bearbeitung.
- Das Hilfe-Element (z.B. Bediener, Werkzeug) hat die gleiche Funktionalität, wird aber einer Maschine zugeordnet.
- Wenn ein Transporter das Wegenetz nicht benutzt, verhält er sich wie eine Maschine und unterscheidet sich lediglich in der Darstellung.

Es gibt drei Elementtypen, mit denen der Transport dargestellt wird.

- Der Transporter ist eine allgemeine Darstellungsweise für den Transport. Wie bereits oben beschrieben, verhält er sich genau wie eine Maschine, wenn er kein Wegenetz benutzt. Wenn er das Wegenetz benutzt, benötigt der Transporter zusätzliche Zeit, um die Entfernungen zurückzulegen, und er kann sich in eine Warteschlange von Transportern einreihen. Der Transporter transportiert periodisch Produkt für Produkt oder Ladung für Ladung.
- Der Förderer ist ein kontinuierlicher Transporter.
- Der Weg ist ein ganz besonderes Element; es führt keine Produkte, sondern Elemente (Transporter). Der Fahrweg verhält sich ähnlich wie ein Förderer. Der Hauptunterschied liegt darin, daß auf einem Förderer das Element bestimmt, wie sich die Produkte verhalten, wohingegen Transporter, wenn sie sich auf einem Fahrweg befinden, ihr Verhalten weitgehend selbst bestimmen können.

Zum Beschreiben der einzelnen Elementtypen gibt es folgende Parameter:

- Elementname für Reporte und Dokumente sowie zur Verwendung in TLI-Ausdrücken
- Elementtyp zum nachträglichen Ändern eines bei der Modellerstellung festgelegten Elementtyps
- Kapazität als Gesamtzahl der Produkte, die sich gleichzeitig in dem Element befinden können
- MTBF (Wert, statistische Verteilung oder TLI-Ausdruck) ist die Zeit von der letzten Reparatur bis zum nächsten Defekt des Elementes (entweder verstrichene Simulationszeit oder tatsächliche Arbeitszeit des Elementes)
- MTTR ist die Zeit, die vergeht, bis ein defektes Element wieder arbeitet
- Fixkosten pro Zeitintervall für jedes Element, egal, ob es gerade arbeitet oder nicht

- Variable Kosten pro Zeitintervall, wenn das Element auch tatsächlich arbeitet
- Zinskosten pro Zeitintervall und Produkt, wenn ein Produkt an dem Element warten muß
- Eingabestrategie (TLI-Ausdruck) zum Regeln des Vorrangs von Produkten, die von verschiedenen Elementen an ein nachgeordnetes Element geschickt werden
- Eingangsbedingung (16 vordefinierte Bedingungen oder TLI-Ausdruck), um die Annahme von Produkten an einem Element einzuschränken
- Ausgangsbedingung (TLI-Ausdruck), so daß Produkte das Element nur dann verlassen können, wenn der TLI-Ausdruck wahr ist
- Verfügbarkeit mit den Anwendungsbereichen Pause, regelmäßige Wartung, Schicht usw.
- Volle Evaluierung zum vollständigen Durchmustern der Warteschlangen nach Produkten, die weitergeschickt werden können
- Element zum Festlegen, ob das betreffende Element in diesem Experiment überhaupt verfügbar sein soll (so können Experimente mit unterschiedlich vielen Ressourcen durchgeführt werden)
- Rüstzeit (Wert oder TLI-Ausdruck) als zusätzliche Zeit, die den Bearbeitungszeiten der für dieses Element definierten Jobs zugeordnet ist
- Warteschlangendisziplin nach verschiedenen Kriterien zur Regelung der Reihenfolge, mit der Produkte in die Warteschlange des Elements eingeordnet werden
- Attributoperationen
- Separieren
- Wegenetz benutzen, Geschwindigkeit, Parkplatz, Zurückkehren, Senden ab/bis und Annehmen ab/bis als spezielle Transporterparameter
- Länge, Fahrtrichtung, Zweispurig, Verdrängen und Geschwindigkeit als spezielle Wegparameter
- Geschwindigkeit, Länge, Abstand der Produkte und Akkumulieren als spezielle Fördererparameter
- Plätze horizontal, Plätze vertikal, Stellplatzgröße und Suche nach weiteren als spezielle Lagerparameter
- Senden ab/bis und Annehmen ab/bis als spezielle Reservoirparameter

Jobs:

An den Elementen können ein oder mehrere Arbeitsgänge durchgeführt werden. Diese Vorgänge werden Jobs genannt. Dabei handelt es sich um die Beschreibung des an diesem Element stattfindenden Prozesses bzgl. Zeitdauer, Anzahl der zu bearbeitenden Produkte usw. Wenn man ein Element definiert, wird von TAYLOR II automatisch ein Job zugeordnet. Man kann aber später weitere Jobs hinzufügen.

Jobparameter sind:

- Jobname für Reporte und Dokumente sowie zur Verwendung in TLI-Ausdrücken
- Zeit (Wert, statistische Verteilung oder TLI-Ausdruck) zur Beschreibung der notwendigen Zeit, um einen Job auszuführen
- Aufnahmemenge (Wert, statistische Verteilung oder TLI-Ausdruck) als Zahl der Produkte, die in einem Arbeitsgang bearbeitet werden
- Abgabemenge (Wert, statistische Verteilung oder TLI-Ausdruck) als Zahl der Produkte, die ausgestoßen werden, wenn der Bearbeitungsprozeß beendet ist und die aufgenommenen Produkte in neue Produkte transformiert wurden
- Trigger am Eingang als TLI-Ausdruck, der ausgeführt wird, wenn ein Produkt (oder eine Menge) einen Job beginnt
- Trigger am Ausgang als TLI-Ausdruck, der ausgeführt wird, wenn ein Produkt (oder eine Menge) einen Job beendet
- Mehrwert für jedes Produkt, das an diesem Job bearbeitet wird
- Hilfe als Anzahl und Art der Hilfe-Elemente, die an diesem Job zur Bearbeitung notwendig sind
- Priorität, die unterschiedlich verwendet werden kann
- Ausschuß als Prozentsatz der Produkte, der an diesem Job als Ausschuß anfällt

Routen und Schritte:

Hat man ein Layout mit einer Anzahl von Elementen und den dazugehörigen Jobs erstellt, muß festgelegt werden, wie dieses Layout benutzt werden soll. Dazu werden eine oder mehrere Routen definiert. Jede Route enthält eine bestimmte Anzahl von Schritten. In einem Schritt ist definiert, welcher Job bzw. welches Element hier benutzt wird und welches der nächste Schritt ist.

Routenparameter sind:

- Sende nach (Wert oder komplexer Ausdruck mit Menge, Bedingungen, Prioritäten) zur Angabe des Schrittes, an den die Produkte geschickt werden, wenn sie den momentanen Schritt verlassen
- Nehme von (Wert oder komplexer Ausdruck mit Menge, Bedingungen, Prioritäten) zur Angabe des Schrittes, von dem bei Montagevorgängen jedesmal ein zusätzliches Produkt angefordert wird

Produkte:

Die Produkte laufen über die festgelegten Routen durch das Layout von dem Schritt aus, an dem sie zu Beginn der Simulation platziert wurden. Man kann ein oder mehrere Produkte bzw. Produkttypen festlegen. Beim Definieren einer Route setzt TAYLOR II automatisch ein Produkt an den Anfang der Route (erster Schritt). Man kann aber auch unterschiedliche Mengen verschiedener Produkte an unterschiedlichen Schritten platzieren.

Produktparameter sind:

- Name für Reporte und Dokumente sowie zur Verwendung in TLI-Ausdrücken
- Icon Nr zur Visualisierung des Produktes während der Animation
- Größe wird bei Förderern und Lagern berücksichtigt und kann mit TLI auch manipuliert werden
- Gewicht mit eigenen Anwendungsmöglichkeiten durch TLI
- Routennummer, Schritt, Produktcode und Anzahl zur Beschreibung, welche Produktvorräte zu Beginn der Simulation an welchen Stellen platziert werden

TLI:

TLI ist eine Sprachschnittstelle von TAYLOR II, mit der eigene Ausdrücke, Berechnungen, Abfragen, Zuweisungen, bedingte Abfragen und Zuweisungen, Mehrfachanweisungen, Schleifen, Funktionen, Befehle und Programme geschrieben und in das Modell eingebunden werden können. Da TAYLOR II in BORLAND PASCAL programmiert wurde, ist die TLI-Syntax analog zu PASCAL, so daß ein einfacher, schnell erlernbarer Umgang mit diesem mächtigen Werkzeug möglich ist.

In TLI kann eine große Anzahl von Systemvariablen im Modell abgefragt und manipuliert sowie in Berechnungen und Anweisungen miteinander verknüpft werden. Insgesamt gibt es 56 elementbezogene Variablen, 22 jobbezogene Variablen, 3 routenbezogene Variablen, 9 produktbezogene Variablen, 21 stochastische Variablen sowie eine Reihe von anwenderdefinierbaren, interaktiven, mathematischen, zeitplanbezogenen und anderen Variablen.

Man verfügt so über sämtliche Informationen, die den aktuellen Modellzustand während der Simulation beschreiben. Damit kann auch das Modellverhalten für die kommende Simulationszeit so verändert werden, daß im Prinzip jedes Verhalten des realen Systems im Modell nachvollzogen werden kann.

Neben den bisher erwähnten Ausdrücken und Variablen können weitere Befehle in TLI benutzt werden. Diese insgesamt 39 Befehle benötigt man zum Formatieren von Reporten, zum Ausgeben von TLI-Daten auf Datei oder Drucker, zum Erstellen von selbständig ablaufenden Präsentationen usw.

Zusätzlich bietet TAYLOR II die Möglichkeit, kleine externe Programme, die in jeder beliebigen Programmiersprache geschrieben und compiliert wurden, aufzurufen und auszuführen. Zum Zwecke des Datenaustausches werden von TAYLOR II beim Aufruf des externen Programms zwei Parameter (Segment- und Offset-adresse) übergeben. An dieser Stelle befindet sich im Speicher ein Feld mit 100 Werten, die vom externen Programm ausgelesen und neu beschrieben werden können. Von TAYLOR II aus kann über TLI-Variablen auf diese Daten zugegriffen werden.

5. Zusätzlich untersuchte Simulationssoftware

5.1. FACTOR/AIM

<i>Hersteller:</i>	Pritsker Corporation 8910 Purdue Road Indianapolis, IN 46268 (USA)
<i>Vertreiber:</i>	ExperTeam SimTec Gesellschaft für Unternehmenslogistik und Qualitätssicherung mbH Emil-Figge-Straße 85, 44227 Dortmund
<i>Preis:</i>	keine Angaben
<i>Anwendung:</i>	Produktion, Logistik, Materialfluß
<i>Anwendergruppe:</i>	Planer, Disponenten
<i>Hardware-Anforderungen:</i>	PC i486 DX mit OS/2 2.1 oder höher, 16 MB Hauptspeicher, 15 MB freier Festplattenspeicher + 5 MB pro Projekt, VGA-Grafik, DB2/2 Datenbankmanager, IBM C/2-Compiler (Das System FACTOR, ohne den Aufsatz AIM ist für alle verbreiteten Rech- nerarchitekturen und Betriebssysteme erhältlich.)

Neutrale Kriterien:

Marktpräsenz:

FACTOR/AIM ist weltweit bei 200 Firmen im Einsatz. Mit 4.700 Pritsker-Simulationssoftware-Installationen (SLAM/SLAMSYSTEM u.a.) besteht ein hoher Marktanteil von Produkten dieser renommierten Softwarefirma. Dabei erstreckt sich das Kundenspektrum vor allem über große und mittelständische Firmen aller Branchen (Industrie, Chemie, Elektrotechnik/Elektronik, Speditionen, u.a.). Der modulare Aufbau gestattet eine sehr flexible Preis-Leistungs-Auswahl, so daß auch für kleinere Simulationenaufgaben ökonomisch sinnvolle Modellierungen möglich sind.

Ergonomie:

FACTOR/AIM besitzt eine grafisch-interaktive Benutzeroberfläche mit voller Fenster- und Mausfunktionalität. Die Bedienung genügt dem SAA/CUA-Standard des OS/2-Presentation-Managers. Masken und Menüs ermöglichen den Zugriff auf alle Modellelemente. Es gibt ein umfangreiches Hilfesystem, inklusive On-Line-Hilfe. Die Kommunikationssprache ist durchweg Englisch. Es gibt ein englischsprachiges Handbuch.

Erlernbarkeit:

Durch den geringen Abstraktionsgrad, die umfangreiche On-Line-Hilfe und das mit Beispielen und Lernaufgaben versehene Handbuch ist der Lernaufwand mittelmäßig einzuschätzen. Grundvoraussetzung für die Bedienung ist eine sichere Beherrschung der englischen Sprache. Vorkenntnisse im Bereich der Simulation sind von Vorteil, aber nicht Bedingung. Auch kleine Modelle lassen sich leicht modellieren und simulieren. Das System komplett zu beherrschen erfordert jedoch eine längere Einarbeitungs- und Anwendungsphase. Informatikspezifische Kenntnisse sind höchstens für die Programmierung der offenen C-Schnittstelle erforderlich.

Sicherheitsfunktionen:

Es bestehen keine Sicherheitsfunktionen gegen Raubkopieren, unberechtigtes Nutzen, Löschen oder Verändern des gesamten Softwaresystems oder einzelner Komponenten und Modelle. Lediglich mit OS/2 können die üblichen Zugriffsrechte und Schutzdienste des Betriebssystems genutzt werden.

Informatikspezifische Kenntnisse

Implementationssprache:

FACTOR/AIM ist in der Programmiersprache C geschrieben. Das System ist offen und modular aufgebaut.

Modellierung:

Ereignisorientierte Modellsicht mit diskreter und/oder dynamischer Zeitfortschreibung. Die gesamte Modellierung verläuft auf einem sehr hohen Niveau, dessen zentrales Objekt der "Auftrag" ist. Die Philosophie ist somit weitestgehend an die PPS- und MRP-Welt angepaßt.

Die FACTOR/AIM-Modellkomponenten sind:

Anlagenelemente:

Maschine, Mitarbeiter, Werkzeug, Vorrichtung, Puffer, Ressourcen-Pool, Materiallager, Transporter-System, Fördersystem, Fahrerlose Transportsysteme

Produktion:

Auftrag, Anforderung, Teil, Arbeitsplan,

Arbeitsgänge:

Bearbeiten, Rüsten, Produzieren, Montieren, Transportieren, Auswählen, Prüfen, Materialbestand ändern, Auftrag auslösen, KANBAN, Losgröße ändern, Batchen

Kontrollelemente:

Schichtpläne, Betriebskalender, Rüstzeitmatrizen, Lookup-Tables

Die Modellierung mit FACTOR/AIM kann über zwei verschiedene Wege erfolgen: zum einen grafisch-interaktiv durch Aufbau eines Flußdiagramm-Netzwerkes oder zweitens durch Datenübernahme aus externen Datenbeständen. Diese beiden Methoden können auch im Verbund angewandt werden.

Durch die offene C-Programmierschnittstelle können neben den parametrierbaren Standardsteuerstrategien beliebige weitere Strategien über Callback-Routinen eingebunden werden. Ausgesprochen fortschrittlich bewerten wir die Modelldatenbank, deren Verwaltung der Datenbankmanager IBM-Extended-Services 2/2 für OS/2 übernimmt. Diese relationale SQL-Datenbank übernimmt nicht nur den Modellimport/-export von bzw. zu anderen Systemen (PPS, BDE, CAD), sondern gestattet auch effiziente und redundanzfreie Modellreplikation und -erweiterbarkeit.

Maximale Modellgröße:

Die statische Modellgröße ist durch die OS/2-Technik des virtuellen Fensters theoretisch unbegrenzt. Praktisch limitiert hier nur Speicher- und Rechenzeitbedarf. Die Anzahl der dynamischen Elemente (Lose) wird auf 5.000 beschränkt. Diese Einschränkung soll im wesentlichen zum Abbruch der Simulation führen, wenn durch einen logischen Modellierungsfehler, der durch die Testalgorithmen nicht erkannt werden kann, die Anzahl der Lose bei einem Simulationslauf unbeschränkt wächst. Sollten 5.000 Lose für ein Modell nicht ausreichen, kann der Maximalwert vergrößert werden.

Statistik:

Über das gesamte Modell und jedes einzelne Modellelement kann eine genaue benutzerdefinierte Statistik geführt werden.

Standardstatistiken sind u.a.:

- Auslastung, Rüstzeit, Bearbeitungszeit, Ausfallzeit etc.
- Warteschlangen-Längen, Wartezeiten
- Durchlaufzeit, Transportzeit, Verspätung, Termineinhaltung
- Materialbestand, Wartezeiten bei Zu- und Abgang
- Streckenstatistiken

Zum Ausblenden der Einschwingphase wird über den Menüpunkt "Clear Statistics-Date/Time" ein Startpunkt für den Beginn der Sammlung von auswertbaren Daten gesetzt. Dieser Zeitpunkt muß vom Nutzer per Experiment oder analytischer Überlegung gefunden werden. Möglich sind auch das Initialisieren des Modells oder das Laden einer zuvor abgebrochenen Simulation.

Zufallszahlengenerierung:

Über den internen Zufallszahlengenerator SN sind 14 verschiedene Verteilungsfunktionen direkt verfügbar. Weitere Verteilungen können unter Nutzung des 0-1-gleichverteilten Zufallszahlenstromes SN mit Hilfe der Programmierschnittstelle hinzugefügt werden. In den Parametermasken für die Attributierung von Bausteinen, Werkstücken, Arbeitsgängen und Kontrollelementen werden immer bereits passende Verteilungsfunktionen vorgeschlagen, die der Nutzer nur noch zu parametrieren braucht.

Robustheit, Verhalten in extremen Situationen

Mit der uns zur Verfügung stehenden DEMO-Version wurden keine Fehlersituationen, Eingabe- und logischen Fehler festgestellt. Die Fehlermeldungen sind genügend präzise und verständlich. Probleme gab es lediglich im Zusammenspiel von Hardware (i386), Betriebssystem OS2/2.1 und FACTOR/AIM. Hier war der leistungs-

schwache Prozessor mit nur 4MB Hauptspeicher eindeutig zu langsam für die praktische Arbeit (Arbeitsunterbrechungen von mehreren Minuten waren die Regel). Auf 16 MB Hauptspeicher (optimal vermutlich 32 MB) sollte bei Neuinstallationen geachtet werden.

Eingabe:

Der Modellentwurf erfolgt grafisch-interaktiv durch Aufbau und Parametrierung eines Flußdiagramm-Netzwerkes. Dabei können beliebige Modellausschnitte vergrößert werden. Das Setzen der Parameter geschieht durch Ausfüllen von Parametermasken.

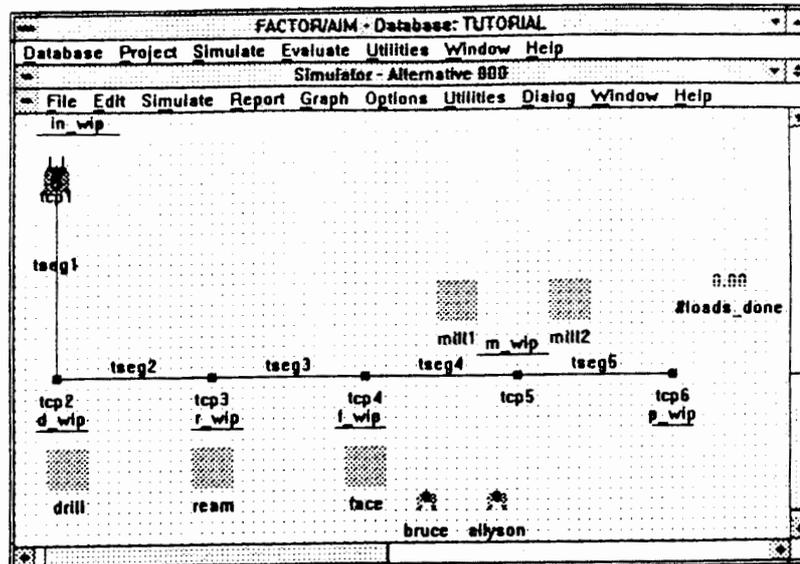
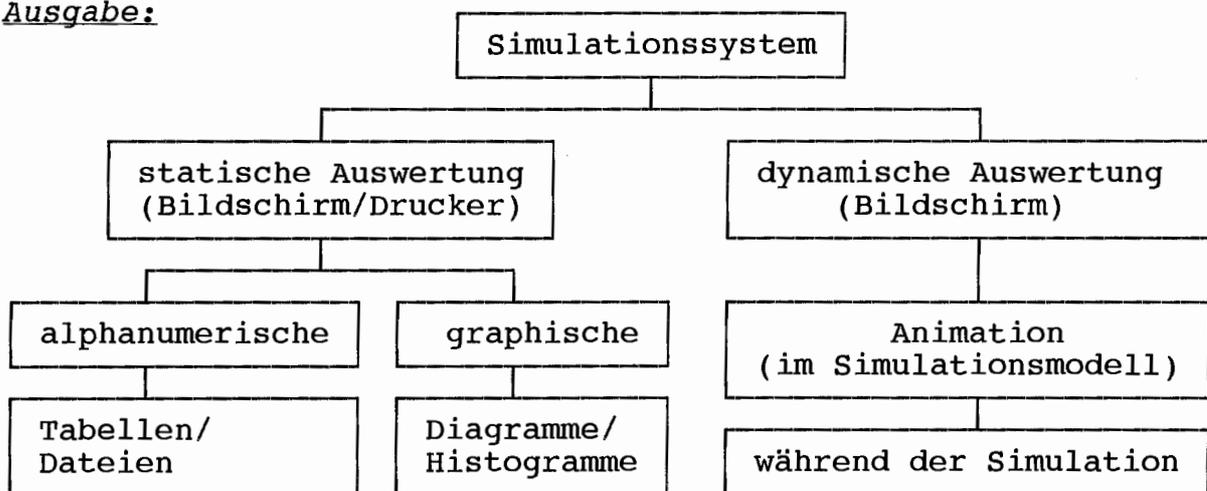


Abb. 5.1 2D-Layout unter FACTOR/AIM [FACT92]

Ausgabe:



Die 2D-Animation wird automatisch mit dem Modell erstellt. Dabei hat der Nutzer umfangreiche Möglichkeiten zur Definition eigener Icons, auch in Hinblick eines dynamischen Produktwandels (vgl. TAYLOR II). 3D-Icons (z.B. von CAD-Systemen) können übernommen werden. Zur Beschleunigung der Simulation läßt sich die

Animation abschalten. Ein Animations-"Film" zur schnelleren Demonstration nach der Simulation kann nicht erstellt werden.

Experimente und Debugging:

Mit der gleichzeitigen Simulation und Animation des Modells steht ein breites Spektrum von Möglichkeiten zum Experimentieren und Analysieren mit dem Modell zur Verfügung. Die Simulation kann jederzeit unterbrochen, gespeichert und zurückgeladen werden. Der genaue Zustand einer Modellkomponente kann jederzeit abgefragt (Snapshot) oder über die Zeit beobachtet (Monitoring) werden. Mit diesen Informationen ist ein beliebiges Eingreifen in die Simulation und Verändern (z.B. Maschinen stilllegen, Mitarbeiter umsetzen, Notfallstrategien testen) gegeben.

Beurteilung des Systems FACTOR/AIM in Hinblick der Nutzung für die Zwecke der Heckert-Chemnitzer Werkzeugmaschinen GmbH

Vorteile:

- niedriger Abstraktionsgrad, es existieren Bausteine aus der realen Welt des Materialflusses;
- gutes Handbuch mit On-Line-Hilfe und Hypertextfunktion;
- niedriger bis mittelmäßiger Einarbeitungsaufwand;
- hohe Flexibilität, d.h. durch entsprechenden Modellier- und Programmieraufwand an jede Materialflußsimulation anpaßbar;
- nutzerdefinierte Daten (z.B. Produktionspläne) einsetzbar;
- sehr gute Modelldatenbank;
- gute und aussagekräftige 2D-Animation;
- Schnittstellen zu CAD, BDE und PPS-Systemen können mit der offenen Programmierschnittstelle realisiert werden;

Nachteile:

- komplettes System sehr groß, mit finanziell hohem Aufwand;
- großer Hauptspeicherbedarf (16-32 MB)
- Nutzer muß sichere Englischkenntnisse besitzen;

5.2. AutoMod

<i>Hersteller:</i>	ASI AutoSimulations Inc. 655 Medical Drive Bountiful, Utah 84010 (USA)
<i>Vertreiber:</i>	Hille GmbH Systemtechnik Friemersheimer Straße 40 47249 Duisburg
<i>Preis:</i>	keine Angabe Campuslizenz 6.200 DM
<i>Anwendung:</i>	beliebig
<i>Anwendergruppe:</i>	Planer, Disponenten
<i>Hardware-Anforderungen:</i>	INDY und INDIGO-Workstations von Silicon Graphics, SUN SPARCstations mit mind. GX Grafik, PC i486 DX 50 (256KB Cache) oder höher IRISVISION-Grafikkarte, mind. 32 MB Hauptspeicher, mind. 200 MB freier Festplatten- speicher, empfohlene Peripherie: HPGL-fähiger A3/A4-Drucker, 17" Monitor mit 1280x1024 Auflösung

Neutrale Kriterien

Marktpräsenz:

Das in den USA entwickelte System wird vorrangig in größeren Unternehmen weltweit erfolgreich eingesetzt. Dabei erstreckt sich das Branchenspektrum über Architektur, Luft- und Raumfahrt, Rohstoffgewinnung, Chemie, Transport und Logistik, Textil-, Nahrungs- und Genußmittelindustrie, Elektrotechnik/Elektronik, Energiewirtschaft, Landwirtschaft, Pharmazie, Verwaltung. Nicht zuletzt wird AutoMod an zahlreichen Ausbildungs- und Forschungseinrichtungen genutzt, z.B. am Institut für Fördertechnik der TU Dresden, dessen Erfahrungen über die Zusammenarbeit mit Herrn Cand.Ing. Zimmermann mit in diese Arbeit einfließen.

Preis und Leistungsvermögen ordnen es in die obere Klasse der Produktionsplanungs- und Simulationssysteme ein. Sein Einsatz dürfte sich erst ab einer gewissen Größe des zu simulierenden Systems amortisieren, wobei allerdings der stark modulare Aufbau eine vielfältige Preis-Leistungs-Kombination gestattet.

Ergonomie:

AutoMod besitzt eine grafisch-interaktive Benutzeroberfläche mit voller Fenster- und Mausfunktionalität. Das System besitzt einen übersichtlichen Menüaufbau und gestattet die Parameter-eingabe über Masken. Es existiert eine On-Line-Hilfe. Die Fehlermeldungen sind präzise und ausreichend. Die Kommunikations-

sprache ist durchweg Englisch. Es gibt ein englischsprachiges Handbuch.

Erlernbarkeit:

Um alle Möglichkeiten des umfangreichen Programmpaketes ausschöpfen zu können, ist ein hoher Lernaufwand erforderlich. Die Grundlagen der Systembedienung sind jedoch schnell erlernbar und führen mit der Beschränkung auf kleine Modelle und wenige Programmmodule zu einem raschen Lernerfolg. Für die Modellabbildung ist die Arbeit in verschiedenen Abstraktionsebenen erforderlich: Während die Materialflußtechnik sehr realitätsnah in einem Flußdiagramm-Netzwerk erfolgt, muß die Steuerlogik über die Programmierschnittstelle mit der Simulationssprache GPSS eingegeben werden. Informatikspezifische Kenntnisse sind zwar keine zwingende Bedingung, aber von großem Nutzen, soll die Leistungsfähigkeit von AutoMod voll ausgeschöpft werden. Der Einsatz kann nur empfohlen werden, wenn ein vom Vertreiber geschulter Mitarbeiter (im Preis inbegriffen) sich regelmäßig mit dem System beschäftigen kann.

Sicherheitsfunktionen:

Die Software ist durch eine Identifikationsnummer und einen Softwarekey gegen Raubkopieren, unberechtigtes Nutzen, Löschen oder Verändern geschützt. Zeitgetaktete Sicherungen und Notsicherungsfunktionen bei Systemabsturz existieren nicht.

Informatikspezifische Kriterien

Implementationssprache:

AutoMod ist in der Programmiersprache C geschrieben. Das System ist offen und stark modular aufgebaut.

Modellierung:

Ereignisorientierte Modellsicht mit diskreter und/oder kontinuierlicher Zeitfortschreibung. Für die Modellierung stehen folgende realitätsnahe Bausteintypen zur Verfügung:

Fördererelemente:

- Hochregal, Bodenförderer, Hängeförderer, FTS, Brückenkran

Kinematikelemente:

- Roboter, Menschen

Statikelemente, Ressourcen, Puffer

Alle Elemente können menügesteuert oder mittels Programmierung parametrisiert werden. Über das Prozeßsystem auf der Programmier-ebene bestehen umfassende Möglichkeiten, sowohl lokale als auch globale Steuerstrategien einzugeben. Zusätzlich besteht die Möglichkeit, die Bearbeitungsaufträge sowohl bei Prozeßbeginn als auch während des Prozeßverlaufs zu generieren.

Maximale Modellgröße:

Durch das Bausteinkonzept ist die Modellgröße theoretisch unbegrenzt. Praktisch limitiert hier nur Speicher- und Rechenzeitbedarf. Dem Detaillierungsgrad der Modellierung und Darstellung sind ebenfalls keine Grenzen gesetzt.

Statistik:

AutoMod bietet standardmäßig eine Vielzahl von statistischen Verfahren an (z.B. Verrechnung von MTBF und MTR-Raten). So können über das gesamte Modell und jedes Modellelement genaue Statistiken geführt werden. Sollten diese nicht ausreichen, können beliebige, benutzerdefinierte Statistiken programmiert werden. Der Modul "AutoStat" ermittelt automatisch die Hochlaufphase (Einschwingsphase) eines Modells und eliminiert auf Wunsch diese Daten aus den Statistikdatenbanken.

Nutzerdefinierte Daten (z.B. Arbeitspläne) sind über Spreadsheets und die Programmierschnittstelle (GPSS) für das System nutzbar zu machen.

Zufallszahlengenerierung:

Über zwei interne Zufallszahlengeneratoren sind acht Verteilungsfunktionen direkt verfügbar. Weitere Verteilungen können unter Nutzung der beiden Generatoren mit Hilfe der Programmierschnittstelle hinzugefügt werden.

Robustheit, Verhalten in extremen Situationen:

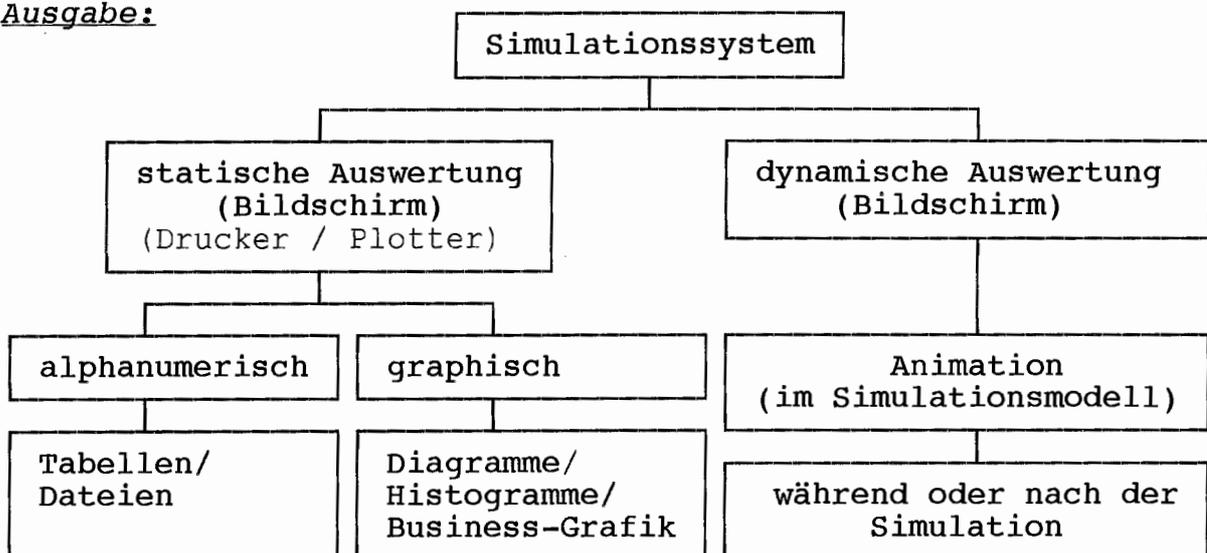
Das gesamte System ist gegenüber Fehlersituationen, Eingabe- und logischen Fehlern sehr sicher. Die Fehlermeldungen sind genügend präzise und verständlich. Im gesamten Testverlauf trat kein undefinierter Zustand, Programmablauf oder -absturz ein.

Eingabe:

Der Modellentwurf erfolgt grafisch-interaktiv durch Aufbau und Parametrierung eines Flußdiagramm-Netzwerkes. Dabei können beliebige Modellausschnitte vergrößert werden. Das Setzen der Parameter geschieht durch das Ausfüllen von Parametermasken, kann aber auch programmiert werden. Dem so entstandenen Layout wird über die GPSS-Programmierschnittstelle die Steuerlogik (Prozeßebene) hinterlegt.

Der Änderungsdienst bereitet keine Schwierigkeiten, erfordert aber den nötigen Überblick über die notwendig werdenden Änderungen auf Prozeßebene.

Ausgabe:



Die 2D/3D-Animation wird automatisch mit dem Modell erstellt. Dabei hat der Nutzer vielfältige Möglichkeiten einer spezi-fischen Definition der Icons für die Systemelemente. Hier kann auch die IGES-Schnittstelle zu CAD-Systemen genutzt werden (z.B. zur Übernahme eines Hallenlayouts). Insbesondere der dynamisch darstellbare Produktwandel (dynamische Icons) ist hervorzuheben. Durch beliebig einstellbare Blickwinkel und Bildausschnitte können die besonders interessierenden Teilprozesse on-line gezoomt werden.

Eine Beschleunigung der Simulation kann durch Abschalten der Animation erfolgen.

Die besonders repräsentative Modelldemonstration bietet sich durch das Erstellen eines Animationsfilmes, der nach Ablauf der Simulation erheblich schneller die Simulation veranschaulicht, als das bei einer On-Line-Animation möglich ist.

Experimente und Debugging:

Der Modul "AutoStat" bietet zahlreiche Möglichkeiten zur umfassenden Darstellung von Abhängigkeiten einzelner Parameter vom Gesamtmodell. Diese können während der laufenden Simulation dynamisch angezeigt (Monitoring) und gespeichert (Snapshot) werden. Vergleichsläufe lassen sich automatisieren (Batchbetrieb), um die optimale Variante zu finden. Ein Konsistenzcheck ist ebenso wie ein Debugger vorhanden. In Verbindung mit dem Syntaxcheck für die GPSS-Programme und der Animation hat der Nutzer insgesamt eine gute Plausibilitätskontrolle zur Verfügung.

Beurteilung des Systems AutoMod in Hinblick der Nutzung für die Zwecke der Heckert-Chemnitzer Werkzeugmaschinen GmbH

Vorteile:

- niedriger Abstraktionsgrad, es existieren Bausteine aus der realen Welt des Materialflusses;
- hohe Flexibilität, d.h. durch entsprechenden Modellieraufwand an jede Materialflußsimulation anpaßbar;
- nutzerdefinierte Daten (z.B. Arbeitspläne) einsetzbar;
- herausragende, aussagekräftige 2D/3D-Animation mit selbstgestaltbaren Icons;
- dynamische Icondarstellung (Produktwandel);
- Erstellung eines Animationsfilms möglich;
- sehr umfangreiche grafische Auswertemöglichkeiten;
- das Angebot eines "Runtime-Moduls" ermöglicht dem Nutzer, sich Modelle als Dienstleistung erstellen zu lassen, ohne selbst das komplette Simulationssystem erwerben zu müssen;

Nachteile:

- hoher finanzieller Aufwand für Software, die mindestens benötigten Standardmodule (Minimalsystem) kosten ca. 100 TDM;
- die sehr umfangreichen Animationsmöglichkeiten sind nur auf teurer, leistungsfähiger Hardware voll nutzbar;
- ein Mitarbeiter muß sich regelmäßig mit dem System beschäftigen können

6. Gewichtete Bewertung der Simulationssysteme

Abschließend soll eine zusammenfassende, gewichtete Bewertung von 5 Simulationssystemen erfolgen, die uns für die praktische Anwendung am interessantesten erscheinen. Es wird darauf hingewiesen, daß die in Abb. 6.1 getroffene Wichtung allgemeinen Überlegungen entspringt und im konkreten Anwendungsfall anders vorzunehmen ist.

Für das Bewertungskriterium "Runtime-Modul" wurde bei AutoMod deshalb die Einzelwertung 0 vergeben, weil die im Informationsmaterial des Vertreibers gemachten Angaben vom Anwender (Institut für Fördertechnik der TU Dresden) nicht bestätigt werden konnten. Vermutlich muß der Runtime-Modul separat erworben werden, während er bei TAYLOR II im Programmpaket enthalten ist und pro vergebener Simulationsstudie (Kopie) vom Hersteller nur lizenziert werden muß.

Tabelle 3.1: Gewichtete Bewertung der Simulationssysteme

Bewertungskriterium	Wichtung	Wertung für Simulationssystem									
		Mosys		Dosimis-3		Perfact		Taylor II		AutoMod II	
Realitätsnähe der Elementedarstellung	2	1	2	2	4	2	4	3	6	3	6
Animationsmöglichkeiten (2D/3D - Animation, Online/Offline - Animation)	3	1	3	1	3	1	3	1	3	3	9
Runtime-Modul (Modellerstellung als Dienstleistung)	1	0	0	0	0	0	0	3	3	0	0
statistische Auswertung (Tabellen, Dateien, Graphiken)	4	1	4	2	8	2	8	3	12	3	12
Benutzerfreundlichkeit (Dokumentation, Menügestaltung, On-Line-Hilfe)	3	2	6	2	6	1	3	2	6	2	6
Unterstützung von Experimenten und Debugging	3	2	6	1	3	1	3	2	6	3	9
Vertretbarkeit des finanziellen Aufwandes (Kosten des Systems)	5	1	5	1	5	1	5	3	15	1	5
gewichtetes Gesamturteil		26		29		26		51		47	

Bedeutung der Einzelwertungen: 0 nicht vorhanden 1 ausreichend 2 gut 3 sehr gut

Abb. 6.1 Gewichtete Bewertung der Simulationssysteme [ZIMM94]

7. Literatur- und Quellenverzeichnis

- [AUTO93] AutoMod II Informationsmaterial,
Hille GmbH Systemtechnik, Duisburg 1993
- [BRAC92] Bracht, U.:
Integration der Simulation in die rechnergestützte
Fabrikplanung,
VDI Berichte 1992, Heft 989, Seiten 73-112
- [CARR88] Carrie, A.:
Simulation of manufacturing systems,
Verlag Wiley & Sons Ltd., Chichester 1988
- [DIOG92] DIOGENES Benutzerhandbuch, Version 6.0
GSI Gesellschaft für Steuerungs- und Informations-
systeme mbH, Berlin 1992
- [DOSI93] DOSIMIS-3 Benutzerhandbuch,
SDZ SimulationsDienstleistungsZentrum GmbH,
Dortmund 1993
- [FACT92] FACTOR/AIM Benutzer- und Referenzhandbuch, Ver. 5.2,
Pritsker Corporation, Indianapolis 1992
- [FACT93] FACTOR/AIM - Einführung in ein Arbeitsmodell,
ExpertTeam SimTec GmbH, Duisburg 1993
- [KEIL94] Keil, B.:
Vorlesungsunterlagen Fakultät Maschinenbau, Lehr-
stuhl Arbeitswissenschaft, TU Chemnitz-Zwickau 1994
- [KUEH91] Kuehne, K. und Reinhardt, A.:
Simulation - eine zukunftsweisende Methode für die
Anlagenplanung,
Zeitschrift "Messen, Steuern, Regeln" 1991, Band 34,
Heft 6, Seiten 263-266
- [MERT92] Mertins, K.:
Rechnergestützte Planung von Fertigungssystemen.
Simulation als Werkzeug des Fabrikplaners,
Zeitschrift "Technica", Zürich 1992, Band 41 Heft 13,
Seiten 16-21
- [MOSY93] MOSYS Benutzerhandbuch, Version 4.1,
FhG-IPK Fraunhofer-Institut für Produktionsanlagen
und Konstruktionstechnik, Berlin 1993
- [NOCH91] Noche, B. und Wenzel, S.:
Marktspiegel Simulationstechnik in Produktion und
Logistik,
Verlag TÜV Rheinland, Köln 1991

- [NOCH92] Noche, B.:
Profis planen niemals "ohne" - Simulationssoftware
für Materialflußabläufe,
Zeitschrift "Materialfluß" 1992, Band 23 Heft 1/2,
Seiten 40-44
- [PAPE90] Pape, D.-F.:
Simulation von Material- und Informationsfluß,
Zeitschrift für Logistik 1990, Band 11 Heft 4,
Seiten 58-63
- [PERF93] PERFECT! Benutzerhandbuch, Version 1.0 Rev. 0.2,
FhG-IML Fraunhofer-Institut für Materialfluß und
Logistik, Dortmund 1993
- [RODE90] Rode, M.:
Produktionslogistik: Analyse und Steuerung durch Si-
mulation,
Verlag TÜV Rheinland, Köln 1990
- [SCHA90] Scharf, P. und Spies, W.:
Fabriksimulation. Ergebnisse einer Befragung von An-
wendern,
VDI Zeitschrift 1990, Band 132 Heft 11, Seiten 62-65
- [SCHA92] Scharf, P. und Spies, W.:
Computersimulation hilft bei der Optimierung von
Produktionsanlagen,
VDI Zeitschrift 1992, Band 134 Heft 10, Seiten 72-76
- [SCHM88] Schmidt, B.:
Simulation von Produktionssystemen,
Fachberichte Simulation, Seiten 1-45,
Springer Verlag, Berlin, Heidelberg, New York 1988
- [SLAM92] SLAMSYSTEM Referenz-Handbuch,
Simulationsanwendung Schröder GmbH, Düsseldorf 1992
- [SPIE91] Spies, W.:
Entwicklung einer rechnergestützten Systematik zur
Simulation von Produktionsabläufen,
VDI Fortschrittsberichte 1991, Reihe 2, Seiten 1-132
- [STAN93] Stanek, W. und Jong, H.de.:
Simulation in der Fabrikplanung und Fertigungssteue-
rung,
Zeitschrift für wirtschaftliche Fertigung und Auto-
matisierung ZWF/CIM 1993, Band 88 Heft 5, Seiten
226-228
- [TANI94] Tanir, O. und Sevinc, S.:
Defining Requirements for a Standard Simulation En-
vironment,
Zeitschrift "COMPUTER" 2/1994, Seiten 28-34
- [TAYL93] TAYLOR II Benutzerhandbuch, Version 2.20,
F&H Logistics and Automation B.V., Tilburg 1993

- [WIEN89] Wiendahl, H.-P.:
Simulation in der Produktionsplanung und -steuerung,
Vortragssonderdruck AWF-PPS-Kongreß 1989
- [ZIMM94] Zimmermann, U.:
Vergleichende Bewertung von Simulationssystemen für
die Akquisition im Maschinenbau,
Diplomarbeit, TU Chemnitz-Zwickau 1994