

Robert Meyer

Th.-Müntzer-Weg 16

07743 Jena

Matr.Nr. 240313

merob@web.de

→ bei Fragen zur Simulation

+49-179-1448134



Studienarbeit Robotik

Simulation eines Werkstückträgerumlaufsystems
mit Tecnomatix eM-Plant 7.0

Prof. Dr.-Ing. Jacobs

Vertiefungsrichtung Produktionstechnik und -Management

am

FB Wirtschaftsingenieurwesen FH Jena

bearbeitet von Robert Meyer

WS 2006/07

Abgabe 05.01. 2007

Inhaltsverzeichnis

Inhaltsverzeichnis	I
Abbildungs- und Tabellenverzeichnis	III
1. Aufgabenstellung	1
1.1 Vorgehensweise	3
2. Modellierung in eM-Plant	4
2.1 Grundlagen	4
2.1.1. Häufig verwendete Syntax	4
2.2 Die Klassenbibliothek	5
2.2.1. Die Vererbungsfunktion	6
2.2.2. Ausschluss der Vererbung	6
2.3 Animation der Objekte	7
3. Das Grundmodell	8
3.1.1. Grundlegende Funktionsweise	9
3.2 Bewegliche Elemente (BE)	11
3.2.1. Der Werkstückträger	11
3.2.2. Das Werkstück	12
3.2.3. Benutzerdefinierte Attribute	12
3.3 Die Förderstrecke	13
3.3.1. Materialflussbaustein Förderstrecke	13
3.3.2. Netzwerk Förderstrecke WUT_xy	13
3.4 Die Ecke (Vereinzelr)	14
3.5 Die Quelle	15
3.6 Die Senke	18
3.7 Der in Reihe angeordnete Arbeitsplatz	21
3.8 Die Ausschleusstation	22
3.8.1. Variante 1 – Ausschleusstation ohne Entstauung	23
3.8.2. Variante 2 - Ausschleusstation mit Entstauung	25
3.9 Der Trigger Zaehler	26
3.10 Die Fertigungssteuerung	27
3.10.1. Der Arbeitsplan	27
3.10.2. Die nächste Station	28

3.11	Die Tabelle Auswertung.....	30
3.12	Variablen	31
3.12.1.	Untergrenze	31
3.12.2.	Obergrenze	31
3.12.3.	Maximale Anzahl Werkstückträger (MaxWST)	31
3.12.4.	Anzahl umlaufender Werkstückträger (AnzUmlaufWST).....	32
3.12.5.	Mittlere Durchlaufzeit (MittlDLZ).....	32
3.12.6.	Durchsatz.....	33
3.12.7.	Bandgeschwindigkeit	33
3.12.8.	Bearbeitungszeit	33
3.13	Methoden auf Netzwerkebene.....	34
3.13.1.	Reset	34
3.13.2.	Die Methode eintragenAuswertung	34
3.13.3.	Die Methode Informationsausgabe	36
3.13.4.	Die Methode Parameter.....	37
4.	Ergebnisse der Modelle	38
4.1	Plausibilitätstest.....	38
4.1.1.	Überlauf.....	38
4.1.2.	Engpass.....	38
4.1.3.	Optimale Werkstückträgeranzahl	39
4.1.4.	Durchsatz.....	39
4.1.5.	Störung	40
4.1.6.	Durchlaufzeit	41
4.1.7.	Ergebnis der Plausibilitätstests.....	43
5.	Kritik und Ausblick	44
5.1	Ansätze und Aufbaumöglichkeiten	44
	Literaturverzeichnis.....	V
	Anhang	VI

Abbildungs- und Tabellenverzeichnis

Tabelle 1:	Anforderungsliste	2
Tabelle 2:	Parameter des Grundmodells	2
Tabelle 3:	Bearbeitung in Reihe ungestört	VII
Tabelle 4:	Bearbeitung in Reihe gestört (10%)	VIII
Tabelle 5:	Ausschleusstation Variante 1 ungestört	X
Tabelle 6:	Ausschleusstation Variante 1 gestört (10%)	XI
Tabelle 7:	Ausschleusstation Variante 2 ungestört	XIII
Tabelle 8:	Ausschleusstation Variante 2 gestört (10%)	XIV
Abbildung 1:	aktive und inaktive Vererbungsfunktion von Objekteigenschaften	6
Abbildung 2:	Modell Umlaufsystem in Reihe.....	9
Abbildung 3:	Ablauf einer Simulation	10
Abbildung 4:	Schema eines Werkstückstückträgerumlaufsystems	11
Abbildung 5:	Werkstückträger mit (links) und ohne (rechts)Werkstück	12
Abbildung 6:	benutzerdefinierte Attribute des Werkstücks	12
Abbildung 7:	Netzwerke Förderstrecken (links) und Bausteine (rechts) in der Klassenbibliothek.....	13
Abbildung 8:	Netzwerk Förderstrecke im Beispiel WUT_uo	13
Abbildung 9:	Netzwerk Ecke im Beispiel Ecke_ru.....	14
Abbildung 10:	Das Netzwerk Quelle	15
Abbildung 11:	Methode Werkstückträger (WST) erzeugen	16
Abbildung 12:	Methode Werkstück auf Werkstückträger bringen	16
Abbildung 13:	Methode Neues Ziel zuweisen	17
Abbildung 14:	Methode Eingang_WST_Montage.....	17
Abbildung 15:	Methode Ausgang_WST_Montage.....	18
Abbildung 16:	Netzwerk Senke.....	19
Abbildung 17:	Methode Bearbeitung_Senke	19
Abbildung 18:	Methode Trennen von Werkstück und Werkstückträger	20
Abbildung 19:	Netzwerk Reihe_Bearbeitung	21
Abbildung 20:	Methode SeriellBearbeitung.....	22

Abbildung 21:	Netzwerk Ausschleusstation_Variante_1	23
Abbildung 22:	Methode zurParallelbearbeitung	24
Abbildung 23:	Methode Bearbeitungszeit.....	24
Abbildung 24:	Methode zurParallelbearbeitung mit Ausschleusfunktion	25
Abbildung 25:	Verknüpfung des Triggers mit Methoden	26
Abbildung 26:	Zeitbereich des Triggers.....	27
Abbildung 27:	Arbeitsplan	28
Abbildung 28:	Methode naechsteStation.....	29
Abbildung 29:	Tabelle Auswertung	30
Abbildung 30:	Behandlung Eingabefehler	31
Abbildung 31:	Methode zur Berechnung der mittleren Durchlaufzeit und Durchsatz.....	32
Abbildung 32:	Methode Reset.....	34
Abbildung 33:	Methode eintragenAuswertung	35
Abbildung 34:	Methode Informationsausgabe	36
Abbildung 35:	Die Methode Parameter.....	37
Abbildung 36:	Abhängigkeit von Durchsatz und Anzahl der Werkstückträger (8h-Schicht)..	40
Abbildung 37:	Abhängigkeit von Durchlaufzeit und Werkstückträgeranzahl	42

1. Aufgabenstellung

Im Rahmen dieser Studienarbeit soll mit der Simulationssoftware Tecnomatix eM-Plant ein Werkstückträgerumlaufsystem (WTUS) modelliert werden. Tecnomatix stellt mit der Software eM-Plant ein umfangreiches Tool zur Lösung von Simulationsaufgaben in der Fertigung zu Verfügung.

Doch wozu nun eigentlich die Arbeit?

Das zu erstellende Modell soll Studenten des Fachbereiches Wirtschaftsingenieurwesen zum Aufbau von Grundverständnis für den Materialfluss in Werkstückträgerumlaufsystemen verhelfen, sowie als Grundlage für Demonstrationszwecke dienen.

Was soll im Einzelnen erreicht werden?

Ziel ist es, ein aus Systembausteinen zusammensetzbares und durch Bearbeiter veränderbares Modell zu schaffen. Damit soll

- der Materialfluss im WTUS realistisch¹ nachgebildet,
- Auswirkung von verschiedenen Parametern auf die Durchlaufzeit und Menge gezeigt,
- Abläufe verständlich visualisiert, sowie
- eine Aufzeichnungsfunktion für die Simulationsergebnisse implementiert

werden.

Darüber hinaus möchte der Verfasser mit dem Modell eine Basis für weitere Simulationsaufgaben für Materialfluss-, Montage-, und Fertigungsprozessketten schaffen.

Tabelle 1 zeigt die wesentlichen Anforderungen an das zu entwickelnde Modell:

Nummer	Anforderung	Umsetzung
1.	Robustheit	- Routinen bei Fehleingaben - Testen auf Fehleingaben und Laufzeitfehler
2.	Nutzerinterface/ Visualisierung	- Übersichtliche Anordnung von Objekten - Kommentierung und Beschreibung von Methoden - intuitive Bedienbarkeit - Animation der Objektsymbole (2D) ²
3.	Korrektheit/ Vollständigkeit	- definierte Parameter (siehe Tabelle 2) - Definition der zu modellierenden Bausteine - Plausibilitätstest

¹ realistisch i.d.S. meint nicht originalgetreu, die für den Simulationszweck relevanten Eigenschaften sind zu abstrahieren

² auf eine 3D Visualisierung wird verzichtet

Nummer	Anforderung	Umsetzung
4.	Wartbarkeit/ Veränderbarkeit	- objektorientiertes Programmieren (Klassenbibliothek) - einstellbare Parameter - Vererbung von Attributen und Methoden
5.	Nachvollziehbarkeit	- Anlegen einer Dokumentation - Kommentierung des Quellcodes und der Objekte - Erstellung einer Auswertungstabelle
6.	umfassende Funktionalität	Modellierung von <ul style="list-style-type: none"> - Quelle - Senke - Förderstrecke - Ecken (Vereinzel) - Variablen - Ausschleusplatz - serieller Arbeitsplatz - Werkstück/Werkstückträger - Fertigungssteuerung/Arbeitsplan - Auswertungstabelle
7.	Software	- eM-Plant 7.0.10 Educational

Tabelle 1: Anforderungsliste

Für das Grundmodell werden folgende Parameter angenommen:

Nummer	Parameter/Einflussfaktoren	Größe
1.	Länge der Anlage (Förderstrecke)	16 m
2.	Geschwindigkeit Förderband	0,2 ms ⁻¹
3.	Aufbringen/Entnahme Werkstück	4 s (variabel)
4.	Bearbeitungs-/Montagezeit	10 s (variabel)
5.	Störungen an Bearbeitungsstation	0...100 % (variabel)
6.	Simulationszeit (Schicht)	8 h/Schicht
7.	Anzahl Werkstückträger im System	Variabel
8.	Objektbausteine	siehe Tabelle 1 (Nr. 6)

Tabelle 2: Parameter des Grundmodells

Die oben genannten Parameter beziehen sich lediglich auf das Grundmodell. Insbesondere variabel hinterlegte Größen sind so auszugestalten, dass sie vom Bediener für verschiedene Simulationsläufe ohne Eingriffe in den Methodencode verändert werden können.

1.1 Vorgehensweise

Die Dokumentation gliedert sich drei Teile:

Zunächst sollen in Kapitel 2 einige grundsätzliche Prinzipien im Umgang mit der Simulationssoftware eM-Plant angesprochen werden.

Daran schließt sich die Dokumentation der Grundmodelle, ihrer Objekte und Methoden, beschrieben in Kapitel 3, an.

In Kapitel 4 werden die Ergebnisse verschiedener Parametereinflüsse analysiert und dokumentiert.

Noch ein Hinweis zur Formatkonvention:

Begriffe, deren Bedeutung sowohl in den Handbüchern zu eM-Plant, als auch in der Sofort-Hilfe (F1) weitergehend erklärt ist, werden in fetter Schriftform unterlegt.

Darüber hinaus sind Eigenbezeichnungen, wie Objekt- und Netzwerknamen in kursiver Form gehalten.

Die Modelldatei *Umlaufsystem_Meyer.spp*, diese Dokumentation, sowie Tabellen-Ausdrucke (.txt Dateien) und Objekte sind im Ordner *C:\Temp\Studenten Tecnomatix\Meyer* auf Rechner 5 im Fabrikgestaltungslabor des Fachbereiches WI abgelegt.

Im Verzeichnis *C:\Temp\Studenten Tecnomatix* auf Rechner 5 befinden sich auch Modelle anderer Bearbeiter.

2. Modellierung in eM-Plant

2.1 Grundlagen

Das Modell wurde in der Version eM-Plant 7.0.10 Educational im Labor Fabrikgestaltung an der FH Jena erstellt. UGS-Tecnomatix offeriert unterjährig Updates in verschiedenen Lizenz-Versionen. Empfehlenswert ist das Angebot von UGS, eine kostenfreie Studentenversion zu beziehen³. Die hiermit erstellten Objekte sind zwar unter der Lizenz auf dem FH Rechner nicht lauffähig, die Studentenlizenz bietet jedoch die Möglichkeit für kleinere Übungssimulationen außerhalb der Fachhochschule.

Nützlich für den effizienten Umgang mit eM-Plant ist die Beherrschung objektorientierter Programmierung. Hierzu sei für Einsteiger das Skript „Grundlagen der Programmierung“ aus der RRZN - Schriftenreihe empfohlen⁴.

Objektorientiertes Programmieren bedeutet im Fall von eM-Plant:

- wiederkehrende Objekte in Klassen zu definieren,
- eine verständliche Ordnerstruktur in der Klassenbibliothek anzulegen, und
- die Vererbungsfunktion in den Instanzen zu nutzen.

Tecnomatix eM-Plant bietet durch Verknüpfung von Objekten und Ansteuerung durch Methoden vielfältige Möglichkeiten, einen Materialfluss zu modellieren. Insofern gibt es keine falsche oder richtige Darstellung. Wesentlich ist jedoch, den – mitunter komplexen - Materialfluss überschaubar, d.h. unter Minimierung von Quellcode und frei von logischen Fehlern abzubilden.

2.1.1. Häufig verwendete Syntax

Einige Grundbegriffe und Bezeichner der in eM-Plant verwendeten Simtalk Syntax sollen kurz erläutert werden:

BE	anonymer Bezeichner für ein bewegliches Element (synonym zu @),
@	anonymer Bezeichner für ein bewegliches Element (Werkstück oder Werkstückträger),
?	anonymer Bezeichner für das Objekt, welches mit der entsprechenden Methode verknüpft ist,

³ Bestellung unter www.emplant.de/student/order.htm

⁴ RRZN, Grundlagen der Programmierung

- nf(Zahl) anonymer Bezeichner für den Nachfolger des mit der Methode verknüpften Bausteines. Die Zahl gibt, an welcher Nachfolger (Nummer der Kante zum Nachfolger) gemeint ist.
- vg(Zahl) anonymer Bezeichner für den Vorgänger des mit der Methode verknüpften Bausteines. Die Zahl gibt, an welcher Vorgänger (Nummer der Kante vom Vorgänger) gemeint ist.
- Standort anonymer Bezeichner für die der Methode übergeordnete Ebene,
- current anonymer Bezeichner für die Ebene, in der sich die Methode befindet,
- basis bezeichnet die höchste Ebene in der Klassenbibliothek (rootfolder).

Die Objekte, deren Eigenschaften und Methoden werden in der Syntax durch Punkte voneinander getrennt [Oberobjekt.Unterobjekt.Eigenschaft].

Auf weitere Details zum Einstieg in Tecnomatix eM-Plant⁵, Standardobjekten und Umgang mit der Software soll hier nicht weiter eingegangen werden. Für die Aneignung von Grundlagenwissen sei auf die bei Herrn Gruhn ausliegende, prägnante Studienarbeit von Katja Kolb⁶ verwiesen. Ebenfalls sehr hilfreich sind die Handbücher von Tecnomatix⁷.

2.2 Die Klassenbibliothek

Die vom Autor modellierten Objekte, Methoden und Module sind in der Klassenbibliothek angelegt.

Hierzu wurde der Ordner **Benutzerbausteine** in die Klassenbibliothek eingefügt, welcher die modellierten Objekttypen enthält.

Der Ordner **Modelle** beinhaltet die drei erarbeitenden und in Abschnitt 3 beschriebenen Grundmodelle.

Des Weiteren enthält die Klassenbibliothek alle Standardobjekttypen für den **Material- und Informationsfluss**.

Warum ist die Modellierung der Objekte in der Klassenbibliothek so dringend empfohlen?

Hierfür soll kurz die Vererbungsfunktion erklärt werden.

⁵Tecnomatix wurde 2005 von Unigraphics (www.ugs.com) übernommen; die Produktbezeichnung eM-Plant wurde 2006 für alle Folgeversionen in Plant Simulation umbenannt

⁶ Katja Kolb, Erstellung eines Tutorials für das Simulationstool in eM-Plant 7.0

⁷ Tecnomatix eM-Plant Objects- bzw. Reference Manual

2.2.1. Die Vererbungsfunktion

Die in der Klassenbibliothek enthaltenen Objekttypen werden als sog. *Klassen* bezeichnet. Die hier festgelegten Eigenschaften und Methoden werden von deren *Instanzen* (eine Kopie der Klasse im jeweiligen Modell) – übernommen und vererbt, soweit die Vererbungsfunktion in den Instanzen aktiviert ist.

Warum ist das so wichtig?

Dies hat zwei Gründe: Angenommen, der Simulationsbetreibende möchte einen Parameter im Modell ändern, um dessen Einfluss zu testen. Am Beispiel des Förderbandes würde dies bei der Modellierung als Klasse bedeuten: Eine Änderung der Länge des Bandes in der Klasse hat zur Folge, dass die Längen aller **Kinder** automatisch angepasst werden. So bleibt dem Bediener der Simulation viel manueller Änderungsaufwand erspart.

Zweitens werden Fehlerquellen infolge „Übersehens“ oder „Vergessens“ einzelner Änderungen, insbesondere bei umfangreichen Modellen, nahezu ausgeschlossen.

2.2.2. Ausschluss der Vererbung

Im Bedarfsfall können einzelne Eigenschaften einer Instanz so gestaltet sein, dass eine Änderung in der Klasseneigenschaft nicht auf die Eigenschaft der Instanz wirken soll. Dann kann die Vererbung in der Instanz deaktiviert werden.

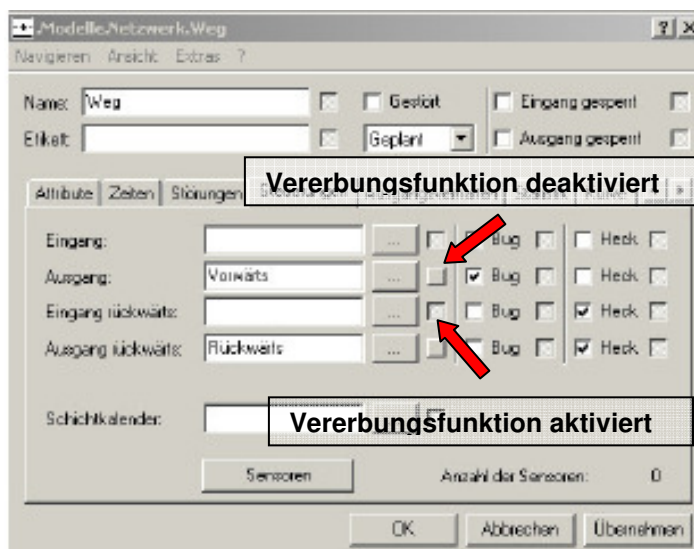


Abbildung 1: aktive und inaktive Vererbungsfunktion von Objekteigenschaften

Bei größeren, häufigeren Abweichungen ist indes zu überlegen, das Klassenobjekt in der Klassenbibliothek zu **duplizieren** und von diesem unabhängigen Objekttyp entsprechende Kinder in das Modell zu implementieren.

Unter Umständen macht es Sinn, **Subklassen** durch **Ableiten** in der Klassenbibliothek anzulegen.

2.3 *Animation der Objekte*

Grundsätzlich erlaubt eM-Plant, sowohl im 2D Modell als auch im 3D Viewer zu modellieren. Dabei ist es möglich, ein 2D-Modell in ein 3D-Modell und vice versa zu übertragen⁸. Entsprechend den Anforderungen wird auf die 3D Modellierung verzichtet.

Vielmehr werden die erstellten Klassen-Module auf der obersten Netzwerkebene, der Ebene des Umlaufsystems zweidimensional animiert.

Die Animation erfolgt über die Funktion **Symbole bearbeiten**:

Zunächst sind Größe und Farbstruktur des Symbols im Zeichenmodus zu bearbeiten. Danach wird der abzubildende Materialfluss im Animationsmodus eingestellt. Hierbei wird der Fluss der sich im Netzwerk befindlichen beweglichen Elemente über **Animationspunkte** bzw. **Linien** der Netzwerkoberfläche zugeordnet⁹.

Die Kanten und Bezeichnungen der Objekte in den Grundmodellen sind zur besseren Übersichtlichkeit ausgeblendet. Sie können über die Menüleiste Ansicht → Optionen → **Kanten**¹⁰ bzw. **Bezeichnung anzeigen** im jeweiligen Netzwerk eingeblendet werden.

⁸ Vgl. hierzu eM-Plant 3D-Viewer, S. 115

⁹ Vgl. eM-Plant Tutorial, S. 72 ff.

¹⁰ Kanten sind Verknüpfungen zwischen Materialflussobjekten, vgl. eM-Plant Hilfe

3. Das Grundmodell

Das Grundmodell ist das jeweils oberste **Netzwerk** „Umlaufsystem“ und besteht aus sieben Modulen, die ihrerseits ein Netzwerk darstellen:

1. bewegliche Elemente (BEs),
2. Förderstrecke,
3. Ecke,
4. Quelle,
5. Senke,
6. Arbeitsplatz in Reihe, sowie
7. Ausschleusarbeitsplatz (Parallel zum Umlaufkreis) in 2 Varianten,

beschrieben in den Abschnitten 3.2 bis 3.8.

Darüber hinaus enthält das Grundmodell Objekte wie die Fertigungssteuerung (incl. Arbeitsplan), eine Tabelle *Auswertung*, den Trigger *Zaehler*, globale Variablen sowie diverse Methoden, welche die einzelnen Objekte verknüpfen und deren Aufruf steuern.

Um die Funktionalität und Anwendbarkeit zu erweitern, wurden drei verschiedene Varianten des Grundmodells erstellt:

1. das *Umlaufsystem_Reihe_Bearbeitung* mit Arbeitsplatz im Umlaufkreis,
2. das *Umlaufsystem_Ausschleus_Var_1* mit Ausschleusplatz ohne Entstauungsfunktion, sowie
3. das *Umlaufsystem_Ausschleus_Var_2* mit Ausschleusplatz und Entstauungsfunktion.

Abbildung 2 zeigt ein beispielhaft das *Umlaufsystem_Reihe_Bearbeitung*.

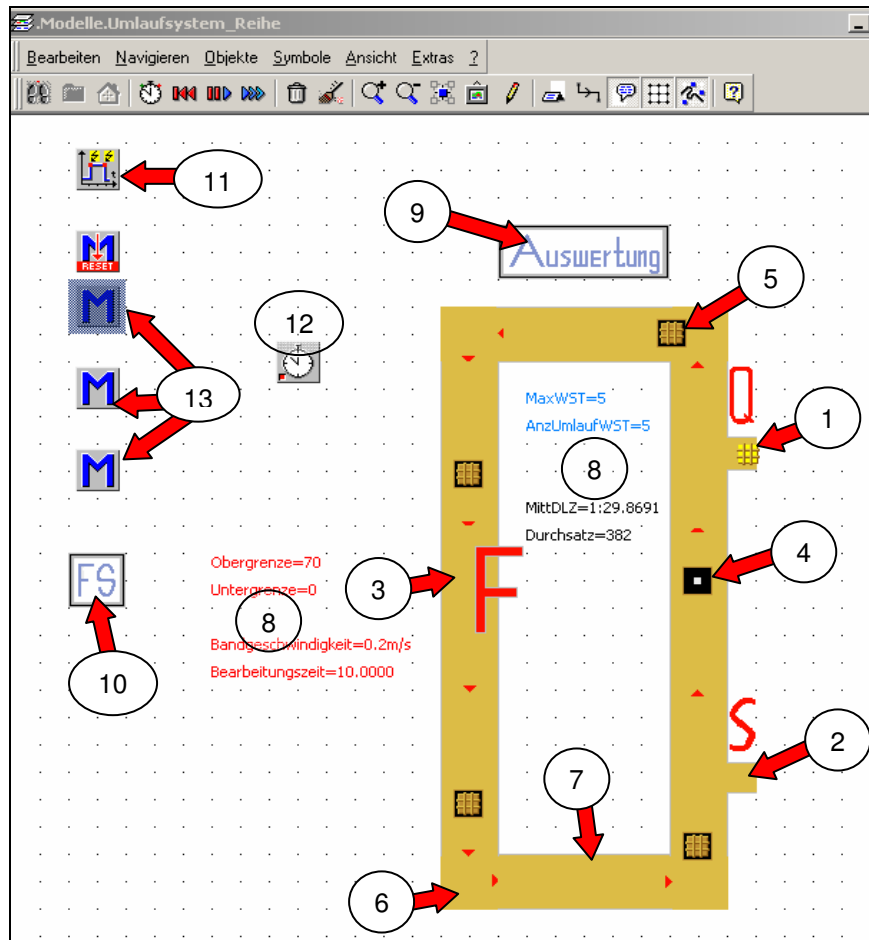


Abbildung 2: Modell Umlaufsystem in Reihe

Letztlich ist ein Grundmodell nichts anderes als die Verknüpfung der Module. Es ist so konzipiert, dass eine Erweiterung des Modells um zusätzliche Bearbeitungsstationen ohne Änderung im Quellcode möglich ist.

3.1.1. Grundlegende Funktionsweise

Wie läuft die Simulation ab?

1. Zunächst baut sich der Bearbeiter aus den Objekttypen der Klassenbibliothek ein Modell zusammen, das seinem Untersuchungszweck entspricht¹¹. Im Ordner Modelle sind bereits drei einfache Grundmodelle angelegt.
2. Die Parameter (Unter-/Obergrenze, Bandgeschwindigkeit, usw.) sind einzugeben.
3. Die Simulation wird im Ereignisverwalter gestartet. In der ersten Schicht werden eine der Untergrenze entsprechende Anzahl Werkstückträger in das Umlaufsystem einge-

¹¹ Beliebig viele Fertigungsstationen, Länge der Förderstrecken, einen Arbeitsplan, Festlegung der Parameter, Verbinden der Module durch Kanten

schleust und mit Werkstücken belegt. Den Aufbau eines WTUS zeigt Abbildung 4 schematisch. Eine Schicht dauert in der Grundeinstellung¹² acht Stunden.

4. In der nächsten Schicht wird ein weiterer Werkstückträger in das Umlaufsystem eingeschleust.
5. Der Bearbeitungsprozess läuft weiter¹³. Am Ende der Schicht werden die Ergebnisse in die Auswertungstabelle geschrieben. Diese Schleife wird solange fortgesetzt, bis die Obergrenze erreicht oder das Umlaufsystem durch die maximale Anzahl Werkstückträger blockiert ist.
6. Alsdann kann die Auswertungstabelle in Excelformat übertragen und analysiert werden und ggf. ein neuer Simulationslauf - mit optimierten Parametern - gestartet werden.

Abbildung 3 zeigt den schematischen Ablauf einer Simulation:

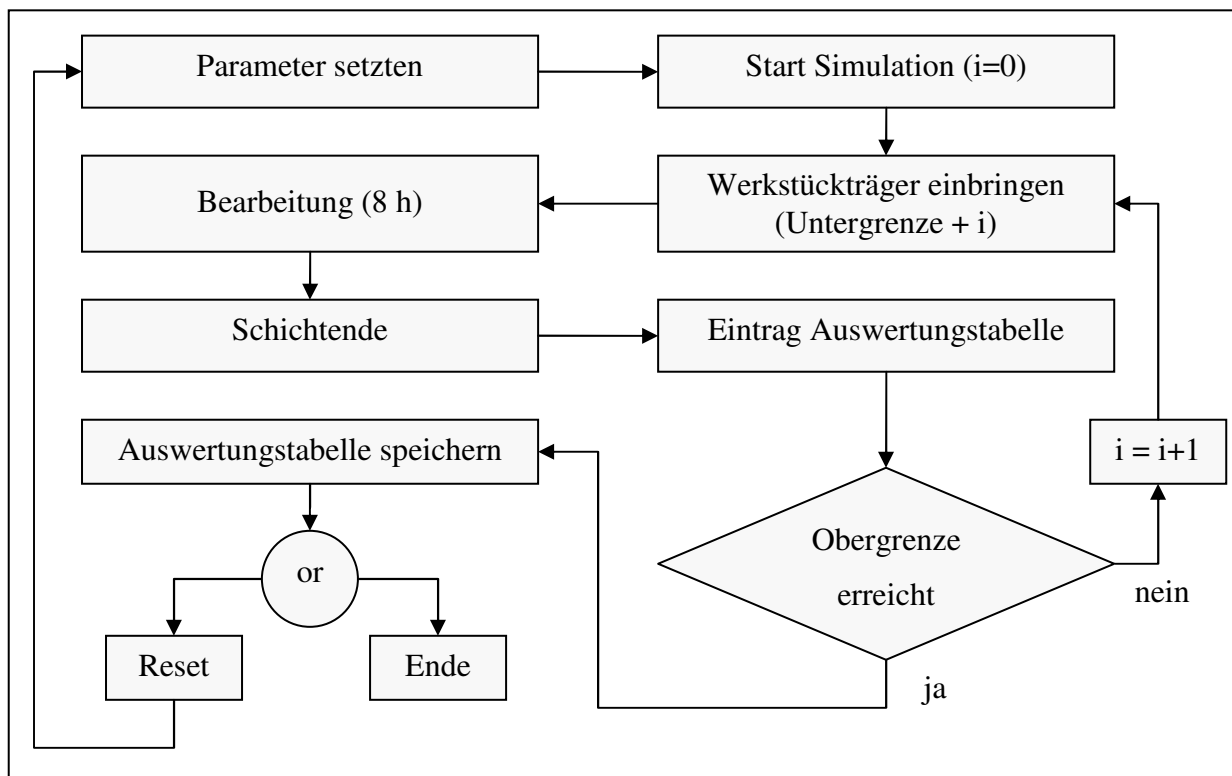


Abbildung 3: Ablauf einer Simulation

¹² die Dauer ist im Schicht-Trigger einstellbar

¹³ analog zur Fließfertigung, wobei die Stationen eines Systems bei Schichtende simultan gestoppt und bei Schichtanfang zeitgleich hochgefahren werden

Das simulierte Werkstückträgerumlaufsystem orientiert sich an einer sog. Stein-Anlage, wie sie sich im Robotiklabor des Fachbereiches Wirtschaftsingenieurwesen an der FH Jena befindet.

Abbildung 4 zeigt den schematischen Aufbau eines solchen Werkstückträgerumlaufsystems:

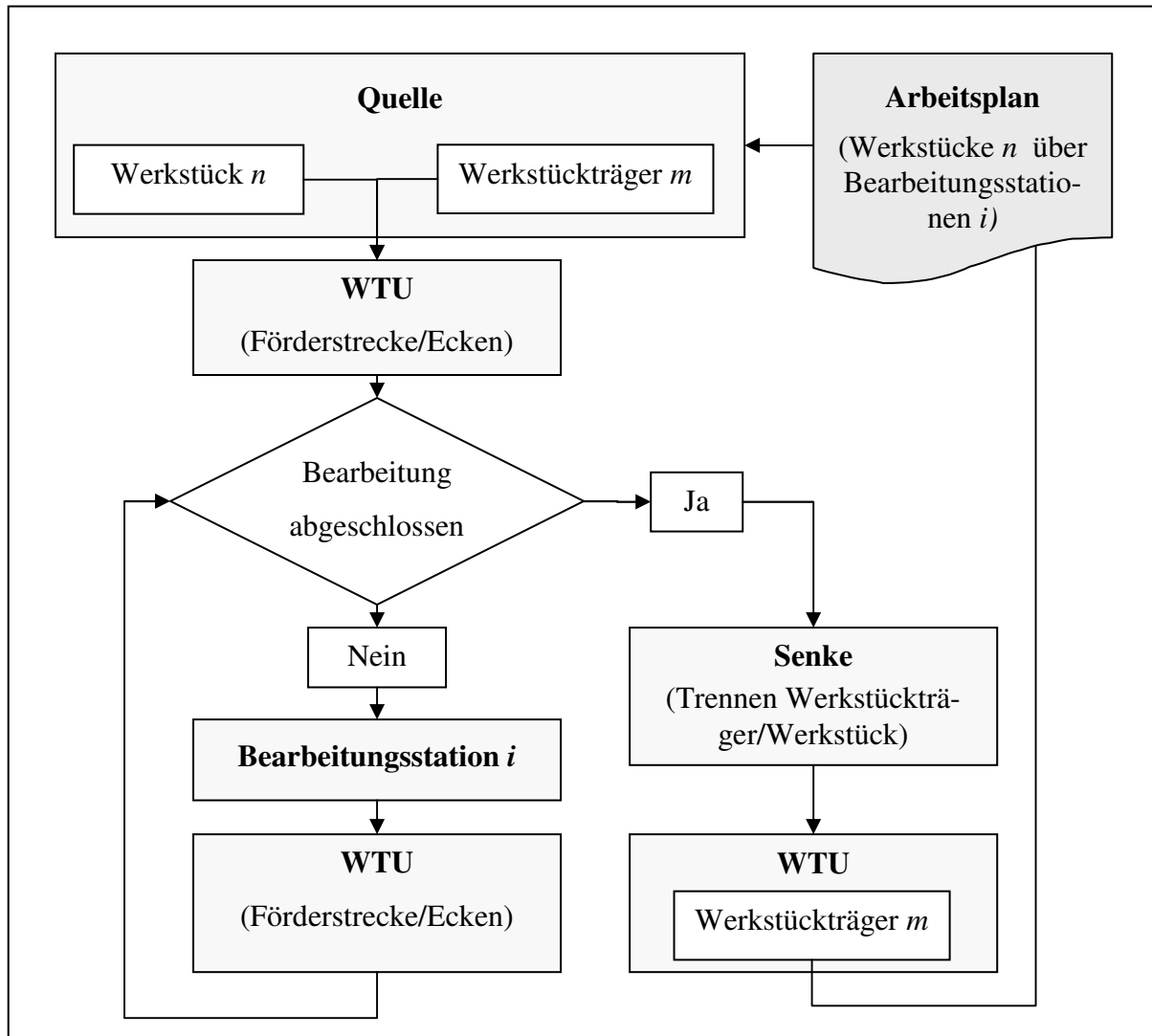


Abbildung 4: Schema eines Werkstückträgerumlaufsystems

3.2 Bewegliche Elemente (BE)

3.2.1. Der Werkstückträger

Der *Werkstückträger* ist vom BE *Förderhilfsmittel* abgeleitet und in der Klassenbibliothek im Ordner BEs hinterlegt. Als solches besitzt er dessen Attribute und Methoden. Ihm wurden eine Länge von 0,3 m sowie ein Buchungspunkt¹⁴ bei 0,15 m Länge zugeordnet.

¹⁴ *Buchungspunkt* ist der Bezugspunkt des Werkstückträgers auf unbeweglichen Materialflussbausteinen.

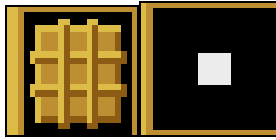


Abbildung 5: Werkstückträger mit (links) und ohne (rechts)Werkstück

3.2.2. Das Werkstück

Das *Werkstück* ist vom BE *Fördergut* abgeleitet und in der Klassenbibliothek im Ordner BEs hinterlegt. Für den geforderten Simulationszweck reicht ein Werkstücktyp aus.

Sollten in weiterführenden Simulationen verschiedene Werkstücktypen benötigt werden, so sind diese vom Objekttyp *Werkstück* abzuleiten.

3.2.3. Benutzerdefinierte Attribute

Dem Objekttyp *Werkstück* wurden zwei zusätzliche Eigenschaften (siehe Registerkarte *Benutzerdefinierte Attribute* in Abbildung 6) angehängt:

1. *BearbSchritt*

Die benutzerdefinierte Eigenschaft *BearbSchritt* ist vom Datentyp *integer* und erfasst die im Arbeitsplan¹⁵ hinterlegten Bearbeitungsschritte. Sie hat den Startwert 1.

2. *Ziel*

Die benutzerdefinierte Eigenschaft *Ziel* ist vom Datentyp *string*. Sie erfasst die im Arbeitsplan hinterlegten Stationsbezeichnungen. Die Eigenschaft *Ziel* wird an jeder Bearbeitungsstation abgefragt und durch die Methode *naechsteStation* auf Übereinstimmung mit dem Stationsnamen geprüft.

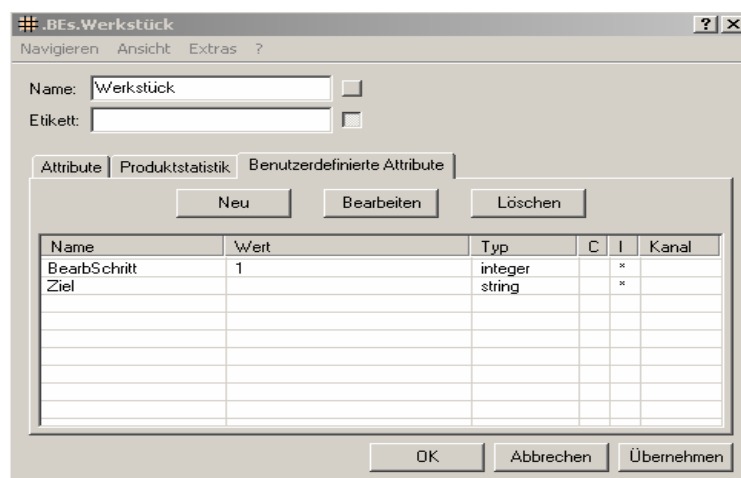


Abbildung 6: benutzerdefinierte Attribute des Werkstücks

¹⁵ siehe auch Abschnitt 3.10.1

3.3 Die Förderstrecke

Die Bezeichnung *Förderstrecke* wird in zwei Verwendungen gebraucht. Zum einen als Materialflussbaustein Förderstrecke, welche in unserer Simulation als *Förderstrecke_1m* sowie *Förderstrecke_2m* in der Klassenbibliothek angelegt ist. Zum anderen bezeichnet Förderstrecke ein Netzwerk.

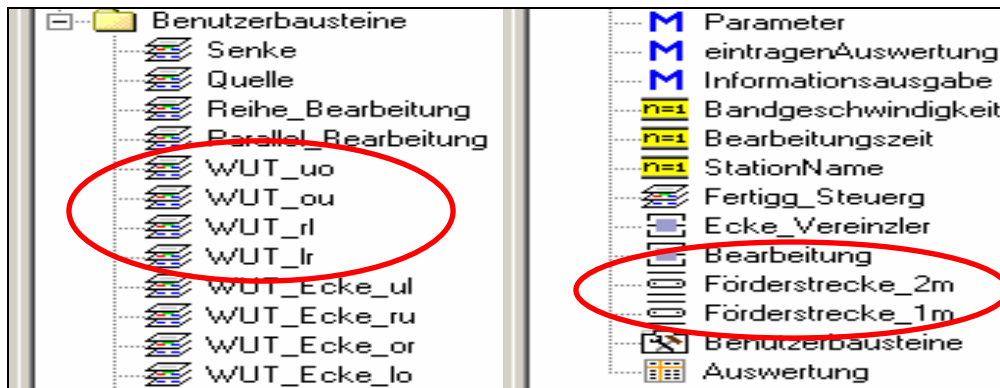


Abbildung 7: Netzwerke Förderstrecken (links) und Bausteine (rechts) in der Klassenbibliothek

3.3.1. Materialflussbaustein Förderstrecke

Der Materialflussbaustein Förderstrecke ist in der Klassenbibliothek mit einer Länge von 2 m bzw. 1 m und einer **Kapazität** von sechs (bei 2 m Länge) bzw. drei Werkstückträgern ausgelegt und als **staufähig** definiert. Mit der Definition der Bausteine in der Klassenbibliothek ist es dem Bearbeiter möglich, Objekteigenschaften wie Länge, Staufähigkeit, Kapazität und Bandgeschwindigkeit¹⁶ für alle Instanzen über Parameter einheitlich und zeiteffizient zu ändern.

3.3.2. Netzwerk Förderstrecke WUT_xy

Das Netzwerk Förderstrecke *WUT_xy* besteht aus dem Materialflussbaustein *Förderstrecke* (hier *Förderstrecke_2m*) sowie je einem Ein- und Ausgang (*Übergang*).

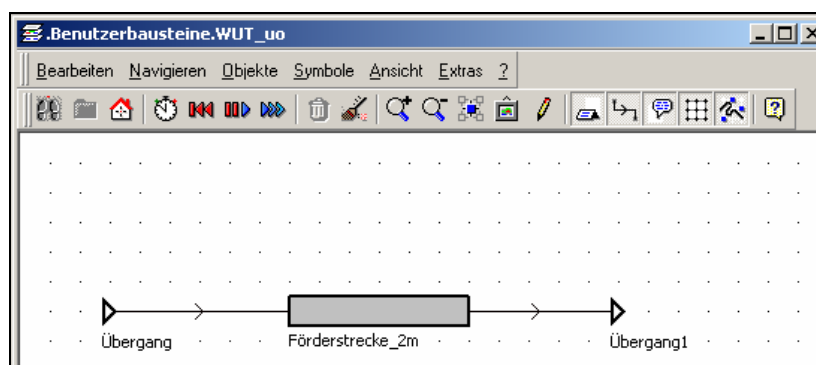


Abbildung 8: Netzwerk Förderstrecke im Beispiel WUT_uo

¹⁶ Bandgeschwindigkeit als Variable; vgl. Abschnitt 3.12.7

Übergänge sind die „Tore“ des Netzwerkes Förderstrecke *WUT_xy* für die Verknüpfung mit anderen Modulen auf der übergeordneten Ebene. Das Netzwerk Förderstrecke ist durch Animationspunkte graphisch animiert (siehe Abschnitt 2.3).

Für die graphische Animation wurden in der Klassenbibliothek *Kinder* des Netzwerkes Förderstrecke *WUT_uo* abgeleitet und in *WUT_xy* umbenannt. Der erste Buchstabe beschreibt dabei die Richtung aus der die Werkstückträger kommen. Der zweite Buchstabe beschreibt, wohin die Werkstückträger fließen (**u**nten, **o**ben, **l**inks, **r**echts). Zur besseren Orientierung deutet ein roter Pfeil die Materialflussrichtung an¹⁷. Die Ableitungen besitzen die Eigenschaften des Elternobjektes, unterscheiden sich jedoch in der graphischen Darstellung. Dem Bearbeiter stehen hiermit fertig animierte, flexibel einsetzbare Materialflussobjekte zu Verfügung.

3.4 Die Ecke (Vereinzler)

Das Netzwerk *WUT_Ecke_xy* besteht aus der Einzelstation *Ecke_Vereinzler* sowie zwei *Übergängen*. Es ist analog zum Netzwerk Förderstrecke *WUT_xy* konzipiert. In der Klassenbibliothek sind entsprechende Ableitungen angelegt.

Die Ecke hat die Aufgabe, die Flussrichtung der Werkstückträger im Werkstückträgerumlaufsystem um jeweils 90° zu ändern, wobei die Werkstückträger den Baustein *Ecke_Vereinzler* durch nur einzeln durchfahren können. Um dies im Modell abzubilden, basiert der Objekttyp *Ecke_Vereinzler* auf dem Objekttyp Einzelstation.

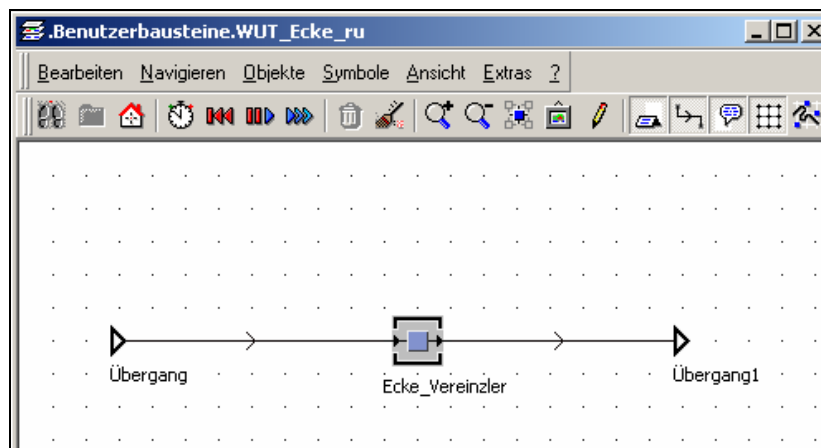


Abbildung 9: Netzwerk Ecke im Beispiel Ecke_ru

Über die Methode *Parameter* wird dem Objekttyp *Ecke_Vereinzler* eine von der Bandgeschwindigkeit sowie Werkstückträgerlänge abhängige Bearbeitungszeit zugewiesen. Die

¹⁷ An dieser Stelle sei daran erinnert, dass die Richtung lediglich für die Animation der graphischen Oberfläche relevant ist. Für die Richtigkeit der Simulation ist die graphische Abbildung hingegen unerheblich.

Bandgeschwindigkeit ist als Parameter vom Bearbeiter global einstellbar (siehe Abschnitt 3.12.7).

3.5 Die Quelle

Die Quelle generiert die beweglichen Elemente (BEs) in einem Umlaufsystem.

Das Netzwerk Quelle besteht aus den Bausteinen *Quelle_WST*, *Quelle_WS*, der Station *WST_Montage* sowie zwei Förderstrecken a 1 m und den obligatorischen Übergängen.

Die Bausteine sind der Darstellung in Abbildung 10 entsprechend mit Kanten verbunden. Die Zahl links vom Zeichen > bezeichnet den Vorgänger (vg) die Zahl rechts vom Zeichen > bezeichnet den Nachfolger (nf) an einer Kante.

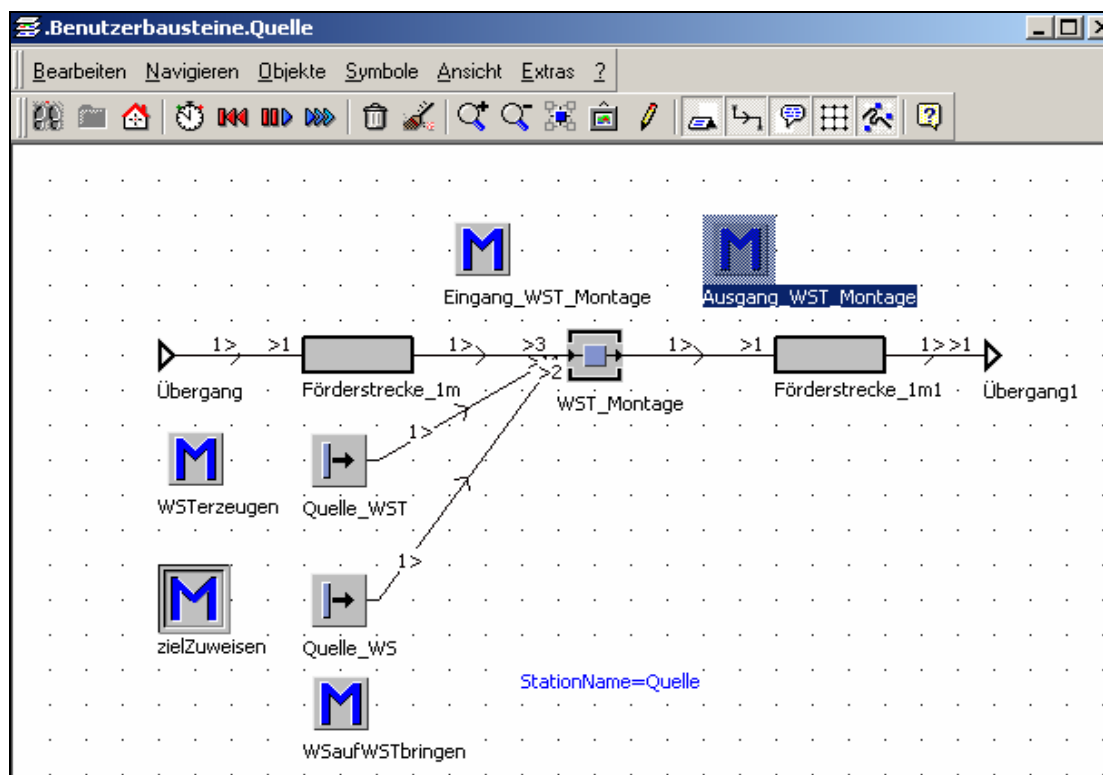


Abbildung 10: Das Netzwerk Quelle

Das Netzwerk *Quelle* belegt die String-Variablen *StationName* mit dem Wert *Quelle*. Die Variable *StationName* ist im Zusammenhang mit der Steuerung des Materialflusses im Arbeitsplan relevant.

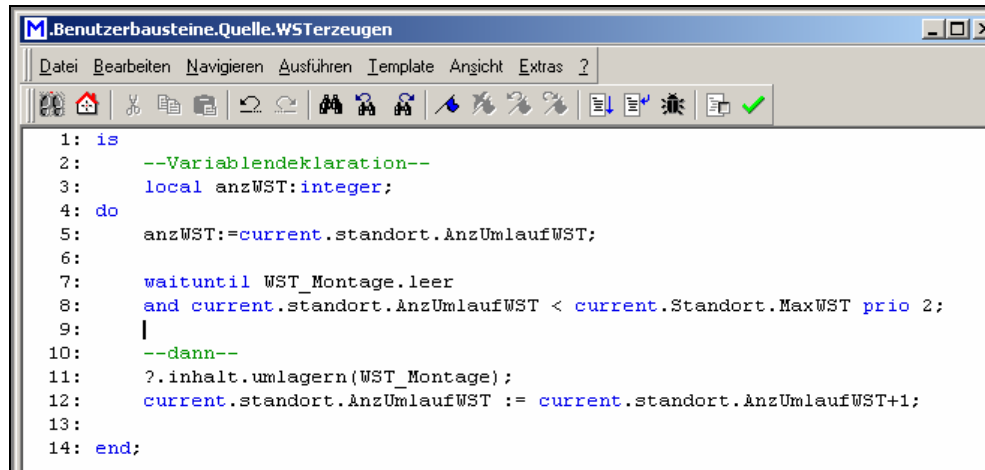
Die Steuerung der Objekte übernehmen die nachfolgend beschriebenen Methoden. Im Baustein *Quelle_WST* werden die Werkstückträger (WST) erzeugt. Die in Abbildung 11 dargestellte Methode *WSTerzeugen* ist mit dem Ausgang dieses Bausteins verknüpft.

Ein Umlagern des – noch leeren – Werkstückträgers vom Baustein *Quelle_WST* in den Baustein *WST_Montage* erfolgt jedoch nur dann, wenn

a) die nachfolgende Station *WST_Montage* nicht belegt ist,

b) die Anzahl der umlaufenden Werkstückträger die Anzahl der im jeweiligen Lauf maximalen Werkstückträger (*MaxWST*) nicht schon erreicht ist.

Durch die **waituntil - prio – Anweisung** wird das Umlagern bis zu dem Zeitpunkt suspendiert, an dem die Ereignisse a) und b) simultan eintreten.



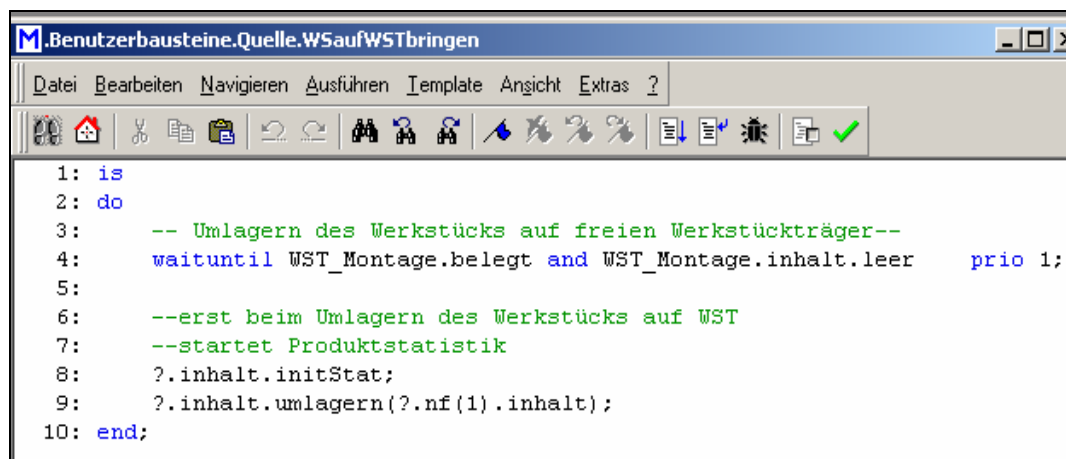
```

1: is
2:   --Variablendeklaration--
3:   local anzWST:integer;
4: do
5:   anzWST:=current.standort.AnzUmlaufWST;
6:
7:   waituntil WST_Montage.leer
8:   and current.standort.AnzUmlaufWST < current.Standort.MaxWST prio 2;
9:   |
10:  --dann--
11:  ?.inhalt.umlagern(WST_Montage);
12:  current.standort.AnzUmlaufWST := current.standort.AnzUmlaufWST+1;
13:
14: end;

```

Abbildung 11: Methode Werkstückträger (WST) erzeugen

Die mit dem Ausgang des Bausteins *Quelle_WS* verknüpfte Methode *WSaufWSTbringen* (siehe Abbildung 12) lagert dann ein Werkstück auf einen in der Station *WST_Montage* befindlichen Werkstückträger um, wenn dieser leer ist. Andernfalls wird mit dem Umlagern gewartet, bis das in Befehlszeile 4 der Abbildung 12 geforderte Ereignis eintritt.



```

1: is
2: do
3:   -- Umlagern des Werkstücks auf freien Werkstückträger--
4:   waituntil WST_Montage.belegt and WST_Montage.inhalt.leer prio 1;
5:
6:   --erst beim Umlagern des Werkstücks auf WST
7:   --startet Produktstatistik
8:   ?.inhalt.initStat;
9:   ?.inhalt.umlagern(?.nf(1).inhalt);
10: end;

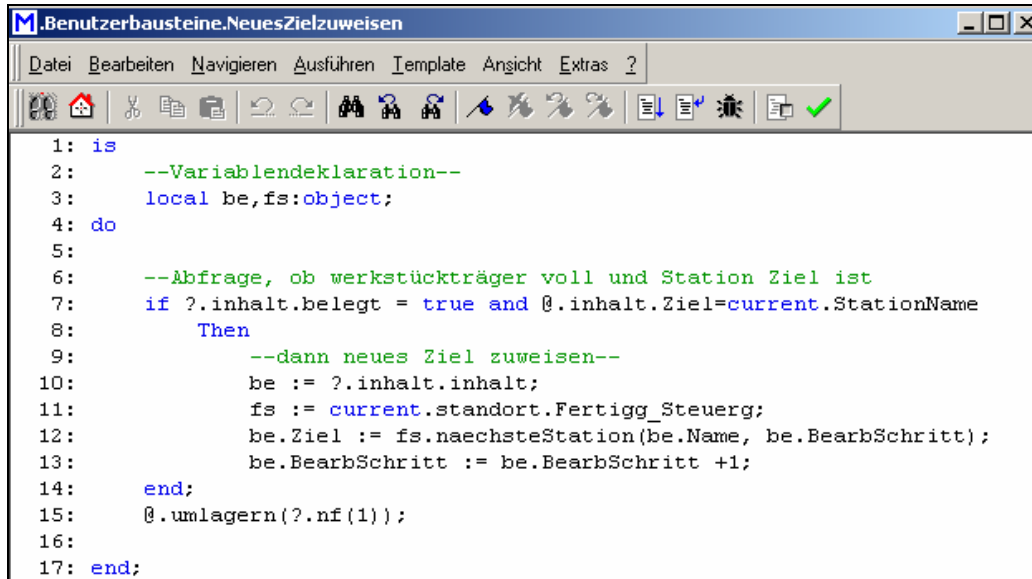
```

Abbildung 12: Methode Werkstück auf Werkstückträger bringen

Zeile 8 der Methode *WSaufWSTbringen* startet beim Umlagern die Zeitstatistik des Werkstücks neu (vgl. Abschnitt 3.12.5).

Die Methode *NeuesZielzuweisen* ist mit dem Ausgang des Bausteins *WS_Quelle* verknüpft und weist dem Werkstück über die Methode *naechsteStation* das nachfolgende Ziel aus der Tabelle *Arbeitsplan* zu (zur Variablenübergabe [*be.Name*, *be.BearbeitSchritt*] siehe Abschnitt 3.10) und erhöht die werkstückeigene Integer-Variable *BearbSchritt* um eins.

Die erste Station im Leben des Werkstücks das Netzwerk *Quelle* selbst (deshalb die Zuweisung *StationName=Quelle* an das Werkstück).

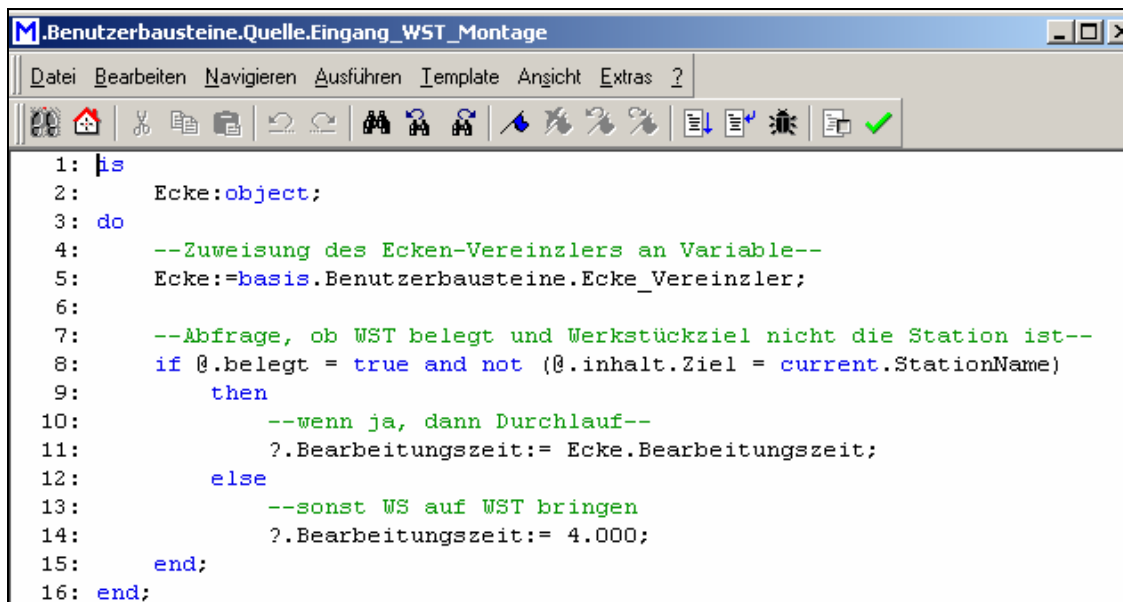


```

1: is
2:     --Variablendeklaration--
3:     local be,fs:object;
4: do
5:
6:     --Abfrage, ob werkstückträger voll und Station Ziel ist
7:     if ?.inhalt.belegt = true and @.inhalt.Ziel=current.StationName
8:         Then
9:             --dann neues Ziel zuweisen--
10:            be := ?.inhalt.inhalt;
11:            fs := current.standort.Fertigg_Steuerg;
12:            be.Ziel := fs.naechsteStation(be.Name, be.BearbSchritt);
13:            be.BearbSchritt := be.BearbSchritt +1;
14:        end;
15:        @.umlagern(?.nf(1));
16:
17: end;
    
```

Abbildung 13: Methode Neues Ziel zuweisen

Die mit dem Eingang des Bausteins *WST_Montage* verknüpfte Methode *Eingang_WST_Montage* fragt ab, ob der eintreffende Werkstückträger belegt ist und falls ja, ob das benutzerdefinierte Attribut *Ziel* des darin enthaltenen Werkstücks mit der Variablen *StationName* übereinstimmt.



```

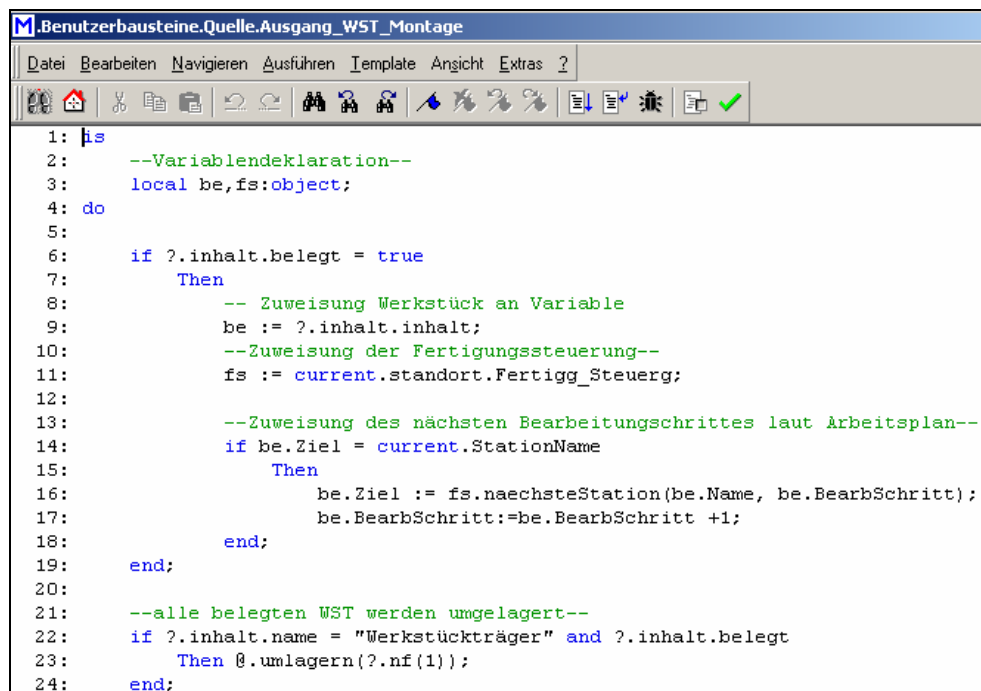
1: is
2:     Ecke:object;
3: do
4:     --Zuweisung des Ecken-Vereinrzlers an Variable--
5:     Ecke:=basis.Benutzerbausteine.Ecke_Vereinrzler;
6:
7:     --Abfrage, ob WST belegt und Werkstückziel nicht die Station ist--
8:     if @.belegt = true and not (@.inhalt.Ziel = current.StationName)
9:         then
10:            --wenn ja, dann Durchlauf--
11:            ?.Bearbeitungszeit:= Ecke.Bearbeitungszeit;
12:        else
13:            --sonst WS auf WST bringen
14:            ?.Bearbeitungszeit:= 4.000;
15:        end;
16: end;
    
```

Abbildung 14: Methode Eingang_WST_Montage

Enthält der Werkstückträger ein Werkstück und ist diesem ein anderes Ziel als das Netzwerk *Quelle* zugewiesen, so soll der Werkstückträger die Station *WST_Montage* mit der dem Objekttyp *Ecke_Vereinrzler* zugewiesenen Bandgeschwindigkeit (vgl. Abschnitt 3.13.4) durchlaufen.

Ist die Bedingung in der Methode *Eingang_WST_Montage* nicht erfüllt, so muss der Werkstückträger logischer Weise leer sein. Dann soll die Aufbringungszeit vom Werkstück auf den Werkstückträger laut Anforderungsliste 4 s betragen¹⁸.

Die mit dem Ausgang des Bausteins *WST_Montage* verknüpfte Methode *Ausgang_WST_Montage* weist dem auf dem Werkstückträger befindlichen Werkstück genau dann ein neues Bearbeitungsziel zu, wenn das Werkstück in der Station *WST_Montage* neu auf den Werkstückträger aufgebracht wurde. Denn nur in diesem Fall wurde dem Werkstück beim Ausgang aus dem Baustein *Quelle_WS* das Ziel *Quelle* zugewiesen.



```
1: fs
2: --Variablendeklaration--
3: local be,fs:object;
4: do
5:
6:   if ?.inhalt.belegt = true
7:     Then
8:       -- Zuweisung Werkstück an Variable
9:       be := ?.inhalt.inhalt;
10:      --Zuweisung der Fertigungssteuerung--
11:      fs := current.standort.Fertigg_Steurg;
12:
13:      --Zuweisung des nächsten Bearbeitungsschrittes laut Arbeitsplan--
14:      if be.Ziel = current.StationName
15:        Then
16:          be.Ziel := fs.naechsteStation(be.Name, be.BearbSchritt);
17:          be.BearbSchritt:=be.BearbSchritt +1;
18:        end;
19:      end;
20:
21:      --alle belegten WST werden umgelagert--
22:      if ?.inhalt.name = "Werkstückträger" and ?.inhalt.belegt
23:        Then @.umlagern(?.nf(1));
24:      end;
```

Abbildung 15: Methode *Ausgang_WST_Montage*

Diejenigen Werkstücke, welche den Baustein *WST_Montage* von seinem Vorgänger Nr. 3 (vgl. Abbildung 10) aus erreichen, sind schon länger im Umlauf und haben ein logischer Weise anderes Ziel - entweder eine Bearbeitungsstation oder die Senke.

Durch die **If-Then – Bedingung** in Befehlszeile 22 der Methode *Ausgang_WST_Montage* ist sichergestellt, dass nur belegte Werkstückträger umgelagert werden.

3.6 Die Senke

Im Materialflusselement Senke verlassen die beweglichen Materialflussobjekte das System. Nachfolgend ist zwischen dem Netzwerk *Senke* und den darin enthaltenen Bausteinen *Senke_WS* und *Senke_WST* zu unterscheiden.

¹⁸ die Montage des Werkstücks auf den Werkstückträger soll während dessen Bewegung durch die Station möglich sein; andernfalls müsste die Bewegungszeit noch hinzuaddiert werden

Das Netzwerk *Senke* besteht aus den Bausteinen *Übergang*, der *Förderstrecke_1m*, den Senken *Senke_WST* und *Senke_WS* sowie der Einzelstation *WS_Senke*. Die Bausteine sind durch Kanten verbunden.

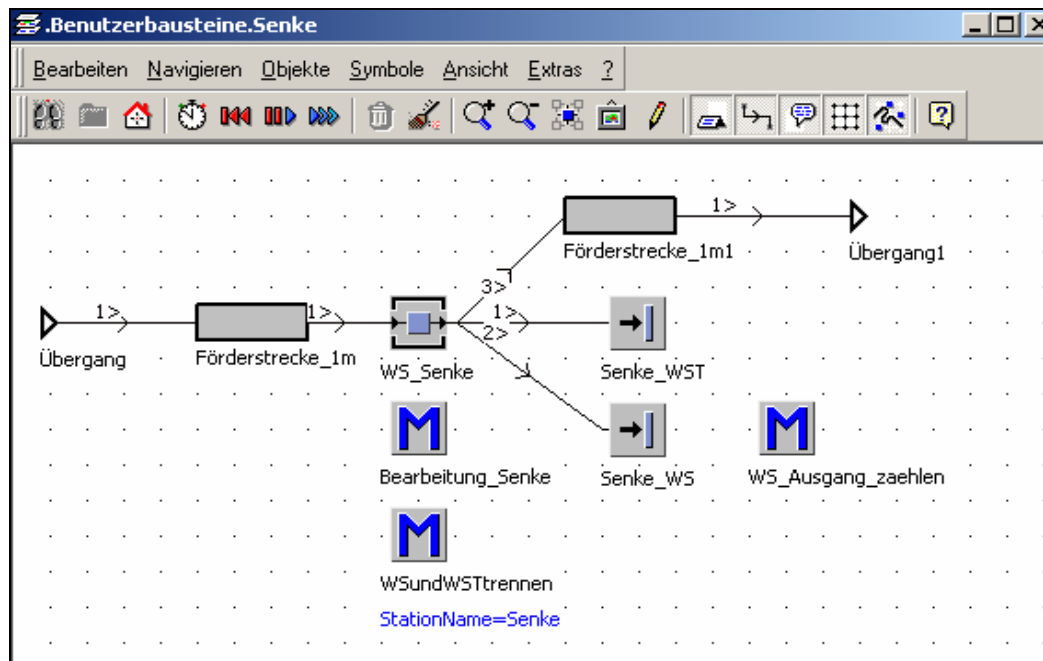


Abbildung 16: Netzwerk *Senke*

Der Materialfluss im Baustein *WS_Senke* wird durch die Methoden *Bearbeitung_Senke* sowie *WSundWSTtrennen* gesteuert. Hierbei ist die Methode *Bearbeitung_Senke* mit dem Eingang, die Methode *WSundWSTtrennen* mit dem Ausgang der Station *WS_Senke* verknüpft.

```

M.Benutzerbausteine.Senke.Bearbeitung_Senke
Datei Bearbeiten Navigieren Ausführen Template Ansicht Extras ?
1: is
2:   --Variablendeklaration--
3:   local WS:object;
4:     Ecke:object;
5: do
6:
7:   Ecke:= basis.Benutzerbausteine.Ecke_Vereinzler;
8:
9:   if @.belegt = true      --Abfrage, ob WST ein Werkstück beinhaltet
10:  then
11:    WS:= @.inhalt;        -- Zuweisung Werkstück an Variable
12:    if WS.Ziel = current.StationName --wenn Senke Werkstückziel ist
13:    then
14:      ?.Bearbeitungszeit:= 4.00; --dann 4s Bearbeitung
15:    else
16:      --sonst Durchlauf mit Geschwindigkeit des Eckbausteins--
17:      ?.Bearbeitungszeit:= Ecke.Bearbeitungszeit ;
18:    end;
19:
20:    -- wenn kein WST kein Werkstück beinhaltet, dann Durchlauf (siehe oben)--
21:  else
22:    ?.Bearbeitungszeit:= Ecke.Bearbeitungszeit;
23:  end;
24: end;

```

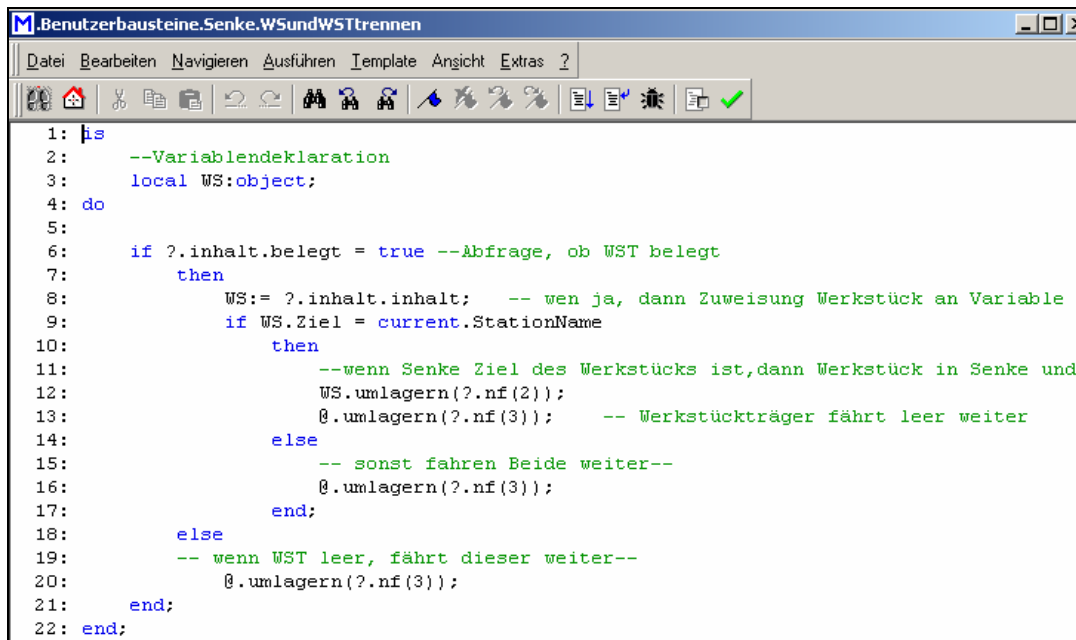
Abbildung 17: Methode *Bearbeitung_Senke*

Die Methode *Bearbeitung_Senke* determiniert die Dauer der Bearbeitungszeit in der Station *WS_Senke*:

Ist das Netzwerk *Senke* nicht Ziel des auf einem Werkstückträger eintreffenden Werkstücks, oder trifft der Werkstückträger leer im Netzwerk *Senke* ein, so durchläuft der Werkstückträger die Station mit der von der Bandgeschwindigkeit abhängigen Zeit¹⁹. Anschließend wird Werkstückträger an den Nachfolger *nf(3)*, die *Förderstrecke_1m*, umgelagert.

Ist die das Netzwerk *Senke* hingegen Ziel des im Werkstückträger enthaltenen Werkstücks, so beträgt die Monatzeit 4 s. Hierzu gilt analog das zur Methode *Eingang_WST_Montage* im Netzwerk *Quelle* Gesagte.

Die Methode *WSundWSTtrennen* bewirkt das Umlagern der Materialflussobjekte am Ausgang der Station *WS_Senke*. Enthält der Werkstückträger ein Werkstück mit dem benutzerdefinierten Attribut *Ziel = Senke*, so wird dieses vom Werkstückträger getrennt und zum Baustein *nf(2) = Senke_WS* umgelagert. Der Werkstückträger wird in jedem Fall – ob leer²⁰ oder belegt²¹ – an den Nachfolger *nf(3) = Förderstrecke_1m* umgelagert.



```
1: is
2:   --Variablendeklaration
3:   local WS:object;
4: do
5:
6:   if ?.inhalt.belegt = true --Abfrage, ob WST belegt
7:     then
8:       WS:= ?.inhalt.inhalt; -- wen ja, dann Zuweisung Werkstück an Variable
9:       if WS.Ziel = current.StationName
10:        then
11:          --wenn Senke Ziel des Werkstücks ist,dann Werkstück in Senke und
12:            WS.umlagern(?.nf(2));
13:            @.umlagern(?.nf(3)); -- Werkstückträger fährt leer weiter
14:          else
15:            -- sonst fahren Beide weiter--
16:            @.umlagern(?.nf(3));
17:          end;
18:        else
19:          -- wenn WST leer, fährt dieser weiter--
20:            @.umlagern(?.nf(3));
21:        end;
22: end;
```

Abbildung 18: Methode Trennen von Werkstück und Werkstückträger

Die mit dem Eingang des Bausteins *Senke_WS* verknüpfte Methode *WS_Ausgang_zaehlen* ist wegen des engen Zusammenhanges mit der *Durchlaufzeit* und dem *Durchsatz* im Abschnitt 3.12.5 beschrieben.

¹⁹ Vgl. Abschnitt 3.13.4

²⁰ der Werkstückträger war schon leer (Fall 1) oder die Senke war Ziel des enthaltenen Werkstückes (Fall 2)

²¹ der Werkstückträger enthält ein Werkstück mit einem anderen Ziel (Fall 3)

3.7 Der in Reihe angeordnete Arbeitsplatz

Im Zentrum des Netzwerkes *Reihe_Bearbeitung* steht die Einzelstation *Bearbeitung*. Die Bausteine sind durch Kanten miteinander verbunden.

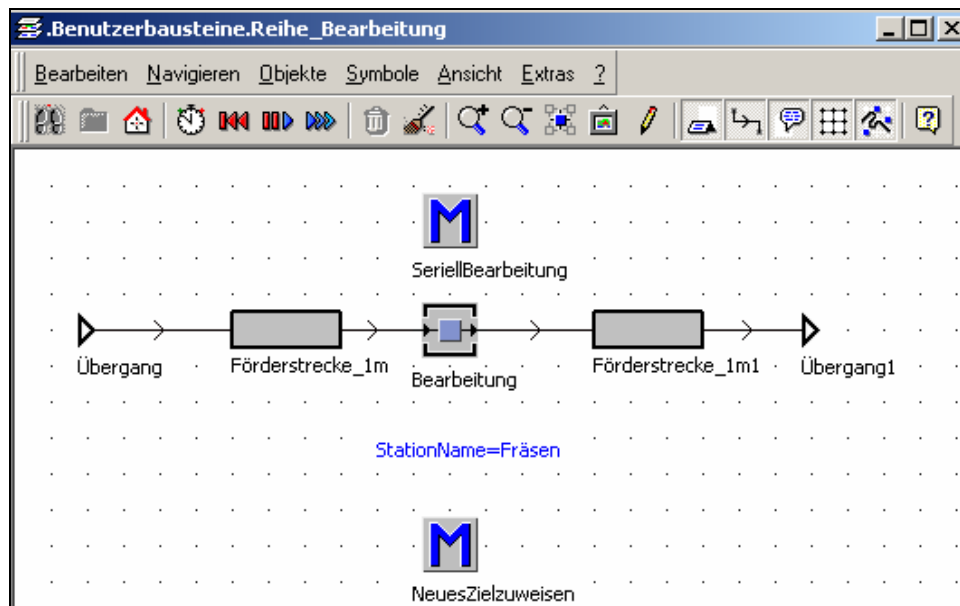


Abbildung 19: Netzwerk *Reihe_Bearbeitung*

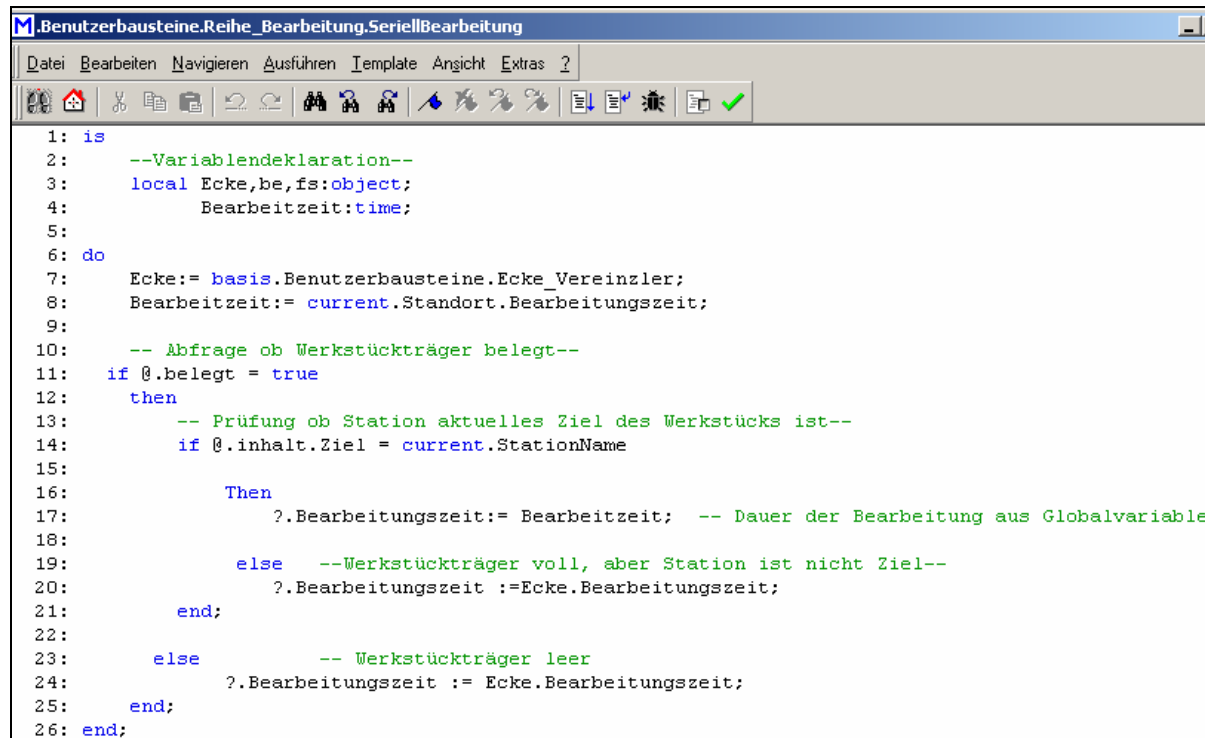
Die mit dem Eingang des Bausteins *Bearbeitung* verknüpfte Methode *SeriiellBearbeitung* weist dem eintreffenden Werkstückträger entsprechende Bearbeitungszeiten zu.

Es werden folgende drei Fälle unterschieden:

1. Ist der Werkstückträger leer, so soll er die Station ohne Bearbeitung entsprechend der eingestellten Bandgeschwindigkeit²² durchfahren (aktuell 3 s).
2. Ist der Werkstückträger belegt und das Netzwerk *Reihe_Bearbeitung* mit der Variable *StationName* Fräsen **nicht** aktuelles Ziel des Werkstücks, so gilt Fall 1.
3. Ist der Werkstückträger belegt und das Netzwerk *Reihe_Bearbeitung* mit der Variable *StationName* Fräsen aktuelles Ziel des Werkstücks, so wird auf die Globalvariable *Bearbeitungszeit*²³ zurückgegriffen (aktuell 10 s).

²² Vgl. Abschnitte 3.12.7 und 3.13.4

²³ Vgl. Abschnitt 3.12.8



```

1: is
2:     --Variablendeklaration--
3:     local Ecke,be,fs:object;
4:         Bearbeitzeit:time;
5:
6: do
7:     Ecke:= basis.Benutzerbausteine.Ecke_Vereinzieler;
8:     Bearbeitzeit:= current.Standort.Bearbeitungszeit;
9:
10:    -- Abfrage ob Werkstückträger belegt--
11:    if @.belegt = true
12:    then
13:        -- Prüfung ob Station aktuelles Ziel des Werkstücks ist--
14:        if @.inhalt.Ziel = current.StationName
15:
16:            Then
17:                ?.Bearbeitungszeit:= Bearbeitzeit; -- Dauer der Bearbeitung aus Globalvariable
18:
19:            else --Werkstückträger voll, aber Station ist nicht Ziel--
20:                ?.Bearbeitungszeit :=Ecke.Bearbeitungszeit;
21:            end;
22:
23:        else -- Werkstückträger leer
24:            ?.Bearbeitungszeit := Ecke.Bearbeitungszeit;
25:        end;
26:    end;

```

Abbildung 20: Methode *SeriellBearbeitung*

Die mit dem Ausgang des Bausteins *Bearbeitung* verknüpfte Methode *NeuesZielzuweisen* ist identisch mit der gleichnamigen Methode im Netzwerk *Quelle*. Insoweit sei auf Abbildung 13 in Abschnitt 3.5 verwiesen.

3.8 Die Ausschleusstation

In Werkstückträgerumlaufsystemen gibt es neben den in Reihe zum Umlaufkreis aufgebauten Bearbeitungsstationen die alternative Anordnung in Form von Ausschleusarbeitsplätzen. Sie sind parallel zum Umlauf angeordnet. Für die Simulation wurden zwei Varianten modelliert:

1. Variante - Ausschleusstation ohne Entstauungsfunktion - kann bei Störung zu Rückstau in den Umlaufkreis führen;
2. Variante - Ausschleusstation mit Entstauungsfunktion – leitet die Werkstückträger bei Überlastung des Eingangspuffers im Umlaufkreis weiter.

3.8.1. Variante 1 – Ausschleusstation ohne Entstauung

Abbildung 21 zeigt den Aufbau des Netzwerkes *Ausschleusstation_Variante_1*:

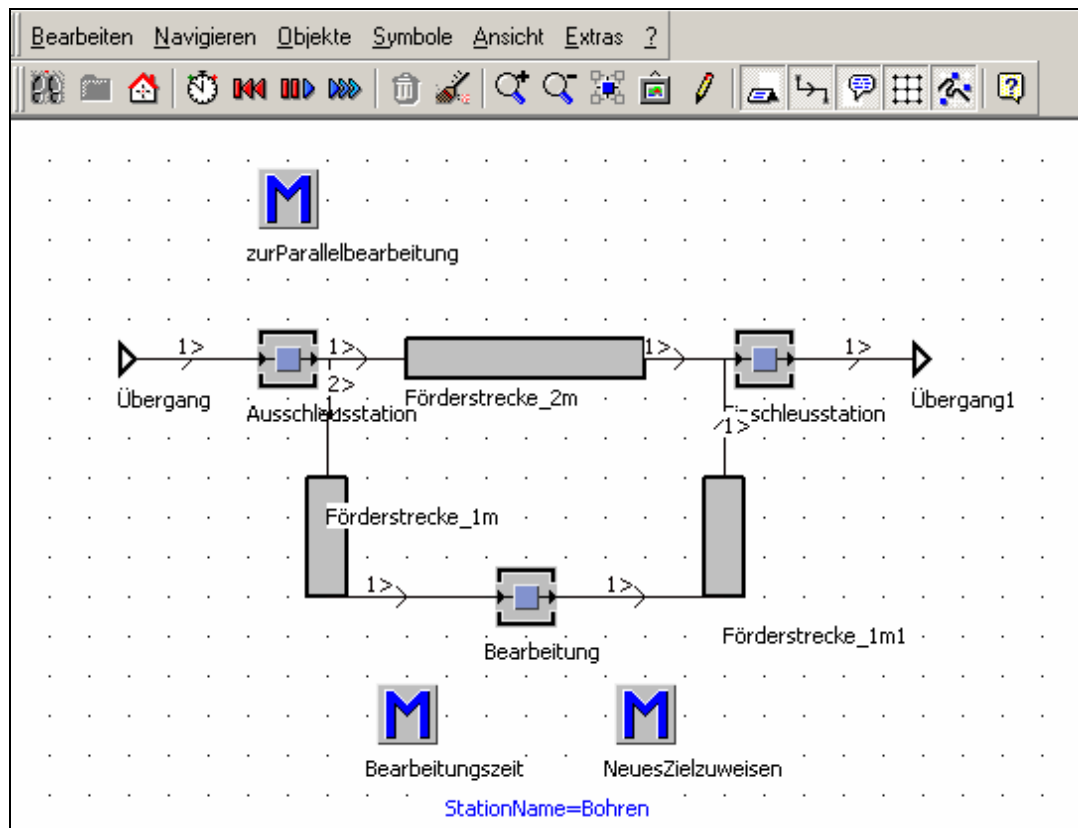
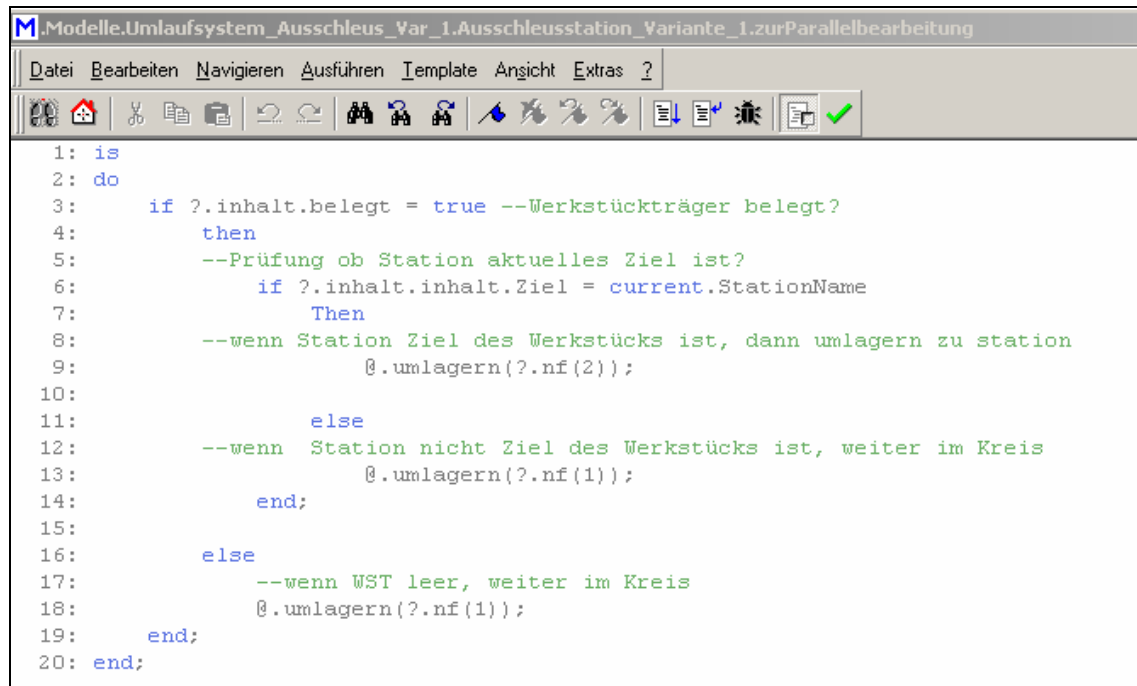


Abbildung 21: Netzwerk *Ausschleusstation_Variante_1*

Das Netzwerk besteht aus den Bausteinen *Ein-* bzw. *Ausschleusstation*, einer den Umlaufkreis darstellenden *Förderstrecke_2m*, der Station *Bearbeitung* sowie den Ein- bzw. Ausgangspuffern *Förderstrecke_1m*.

Die Methode *zurParallelbearbeitung* ist mit dem Ausgang des Bausteins *Ausschleusstation* verknüpft.



```

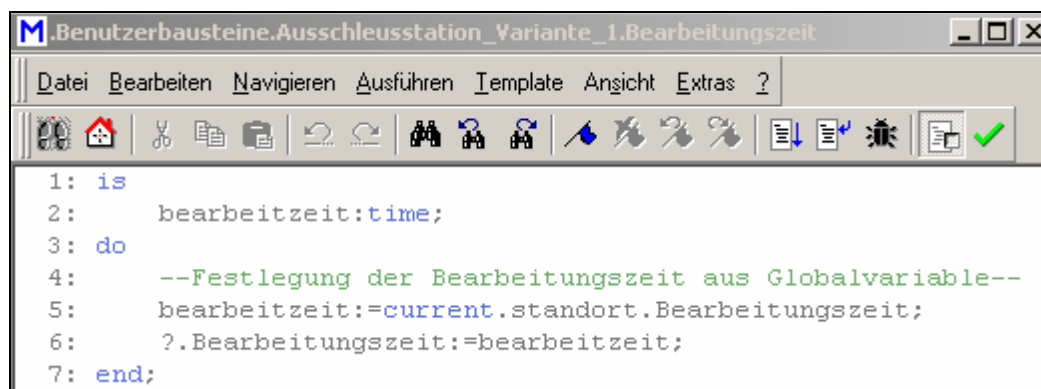
M:\Modelle.Umlaufsystem_Ausschleus_Var_1.Ausschleusstation_Variante_1.zurParallelbearbeitung
Datei Bearbeiten Navigieren Ausführen Template Ansicht Extras ?
1: is
2: do
3:     if ?.inhalt.belegt = true --Werkstückträger belegt?
4:         then
5:             --Prüfung ob Station aktuelles Ziel ist?
6:             if ?.inhalt.inhalt.Ziel = current.StationName
7:                 Then
8:                     --wenn Station Ziel des Werkstücks ist, dann umlagern zu station
9:                     @.umlagern(?.nf(2));
10:
11:                 else
12:                     --wenn Station nicht Ziel des Werkstücks ist, weiter im Kreis
13:                     @.umlagern(?.nf(1));
14:                 end;
15:
16:             else
17:                 --wenn WST leer, weiter im Kreis
18:                 @.umlagern(?.nf(1));
19:             end;
20: end;

```

Abbildung 22: Methode zurParallelbearbeitung

Im Netzwerk *Ausschleusstation_Variante_1* werden alle Werkstückträger, die ein Werkstück mit dem Ziel = *StationName* beinhalten, zur Station *Bearbeitung* ausgeschleust. Dies gilt auch, dann wenn deren Eingangspuffer voll ist. In diesem Fall „wartet“ der Werkstückträger im Baustein *Ausschleusstation*. Im Ergebnis führt die Anordnung im Netzwerk *Ausschleusstation_Variante_1* bei Überlauf des Eingangspuffers *Förderstrecke_1m* – etwa in Folge einer Störung am Baustein *Bearbeitung* – zum Rückstau in den Umlaufkreis.

Die Methode *Bearbeitungszeit* ist mit dem Eingang des Bausteins *Bearbeitung* verknüpft. Sie weist dem Baustein *Bearbeitung* die Bearbeitungszeit entsprechend der Globalvariable *Bearbeitungszeit*²⁴ (10 s) zu.



```

M:\Benutzerbausteine.Ausschleusstation_Variante_1.Bearbeitungszeit
Datei Bearbeiten Navigieren Ausführen Template Ansicht Extras ?
1: is
2:     bearbeitzeit:time;
3: do
4:     --Festlegung der Bearbeitungszeit aus Globalvariable--
5:     bearbeitzeit:=current.standort.Bearbeitungszeit;
6:     ?.Bearbeitungszeit:=bearbeitzeit;
7: end;

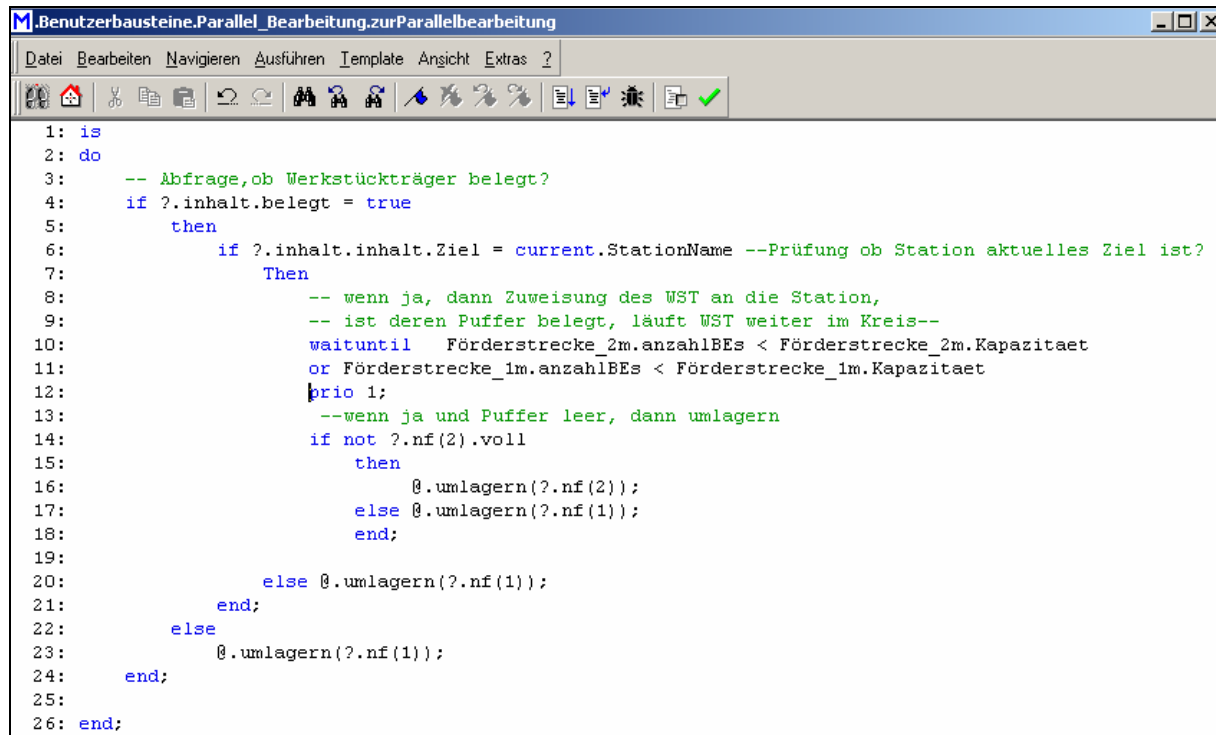
```

Abbildung 23: Methode Bearbeitungszeit

²⁴ siehe Abschnitt 3.12.8

3.8.2. Variante 2 - Ausschleusstation mit Entstaung

Das Netzwerk *Ausschleusstation_Variante_2* im Grundmodell *Umlaufsystem_Ausschleus_Var_2* basiert auf dem in Abschnitt 3.8.1 beschriebenen Netzwerk *Ausschleusstation_Variante_1*. So ist der Aufbau identisch (vgl. Abbildung 21). Die Netzwerke unterscheiden sich lediglich in der mit dem Ausgang des Bausteins *Ecke_Vereinlzler* verknüpften Methode *zurParallelbearbeitung*:



```

1: is
2: do
3:   -- Abfrage, ob Werkstückträger belegt?
4:   if ?.inhalt.belegt = true
5:     then
6:       if ?.inhalt.inhalt.Ziel = current.StationName --Prüfung ob Station aktuelles Ziel ist?
7:         Then
8:           -- wenn ja, dann Zuweisung des WST an die Station,
9:           -- ist deren Puffer belegt, läuft WST weiter im Kreis--
10:          waituntil Förderstrecke_2m.anzahlBES < Förderstrecke_2m.Kapazitaet
11:          or Förderstrecke_1m.anzahlBES < Förderstrecke_1m.Kapazitaet
12:          prio 1;
13:          --wenn ja und Puffer leer, dann umlagern
14:          if not ?.nf(2).voll
15:            then
16:              @.umlagern(?.nf(2));
17:            else @.umlagern(?.nf(1));
18:          end;
19:        else @.umlagern(?.nf(1));
20:      end;
21:    else
22:      @.umlagern(?.nf(1));
23:    end;
24:  end;
25:
26: end;

```

Abbildung 24: Methode zurParallelbearbeitung mit Ausschleusfunktion

Im Gegensatz zur Variante 1 aus Abschnitt 3.8.1 wird nur dann ein Werkstückträger zur Bearbeitungsstation ausgeschleust, wenn

- a) die Ausschleusstation Ziel (hier Bohren) des darin enthaltenen Werkstücks ist, **und**
- b) der Puffer der Ausschleusstation, die *Förderstrecke_1m*, zum Zeitpunkt des Umlagerungsversuchs nicht voll ist.

Sind beide Nachfolger der Ausschleusstation belegt, so wird das Umlagern solange suspendiert, bis die erste der beiden Nachfolger²⁵ frei wird.

Im Code der Methode sind dabei alle denkbaren Fälle berücksichtigt:

1. Ist der Werkstückträger leer, bleibt er im Umlaufsystem.
2. Hat das im Werkstückträger enthaltene Werkstück ein anderes Ziel, bleibt der Werkstückträger im Umlaufkreis.

²⁵ *nf(1)* = *Förderstrecke_2m* → *Umlaufsystem*; *nf(2)* = *Förderstrecke_1m* → *Ausschleusstation*

3. Hat das im Werkstückträger enthaltene Werkstück die Bearbeitungsstation als Ziel, so wird zu dieser umgelagert, wenn ihr Puffer nicht voll ist.
4. Sind beide Nachfolger voll belegt, so wird unter den Voraussetzungen des Falles 3 so lange gewartet, bis der erste der beiden Nachfolger frei wird.

Im Ergebnis führt die Änderung in der Methode *zurParallelbearbeitung* gegenüber Variante 1 dazu, dass ein Rückstau in den Umlaufkreis - etwa bei längeren Störungen an der Ausschleusstation - verhindert wird. Die Werkstückträger fließen weiter im Umlaufkreis, müssen aber die Station, soweit keine Alternativstation im Umlaufkreis existiert, in einer weiteren Runde noch einmal anfahren. Insofern verlängert sich die Durchlaufzeit ebenfalls. Im simulierten Modell kommt es sogar zu erheblichen Schwankungen der Durchlaufzeit bei suboptimaler Werkstückträgeranzahl, wie Abbildung 37 (Seite 42) zeigt. Auf den Durchsatz hat die Änderung gegenüber Variante 1 keine Auswirkung, wie Abbildung 36 zeigt.

Jedoch können in Umlaufsystemen mit mehreren gekoppelten Ausschleusarbeitsplätzen ggf. Werkstückträger, die Werkstücke mit anderen Zielen beinhalten, weiter im System fließen. Ohne Rückstau erreichen sie ihre Zielarbeitsplätze schneller und erhöhen so deren Auslastung und Durchsatz. Hierzu können weiterführende Modellierungen vorgenommen werden, wobei die beschriebenen Netzwerke kombinierbar sind.

3.9 Der Trigger Zaehler

Der Schicht-Trigger *Zaehler* löst in definierten Abständen die mit seiner Registerkarte Aktionen verknüpften Methoden *eintragenAuswertung*, *Informationsübergabe* sowie *Parameter* aus.

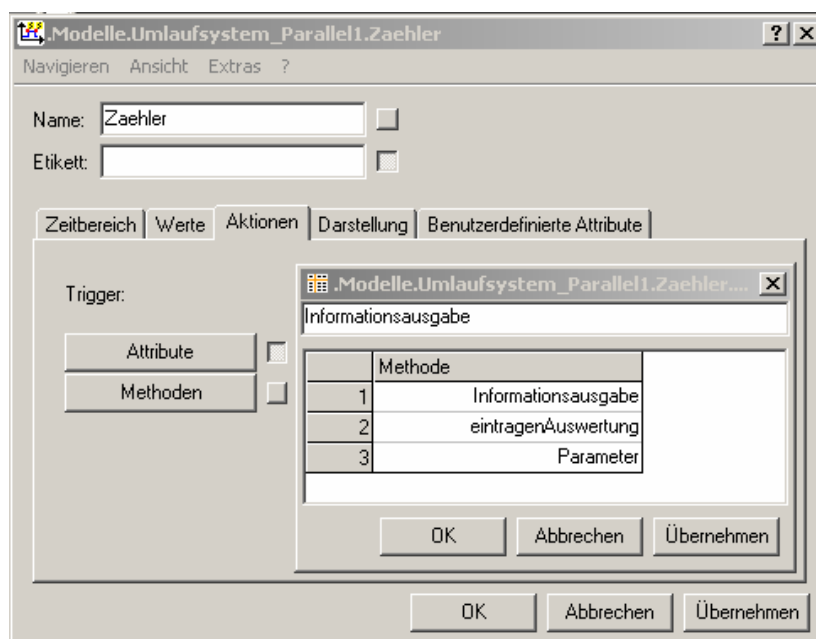


Abbildung 25: Verknüpfung des Triggers mit Methoden

Der Trigger *Zaehler* bestimmt damit die Länge einer Schicht (Zyklus). Die Dauer eines Zyklus ist den Vorgaben entsprechend auf 8 Stunden eingestellt und kann ggf.²⁶ variiert werden.

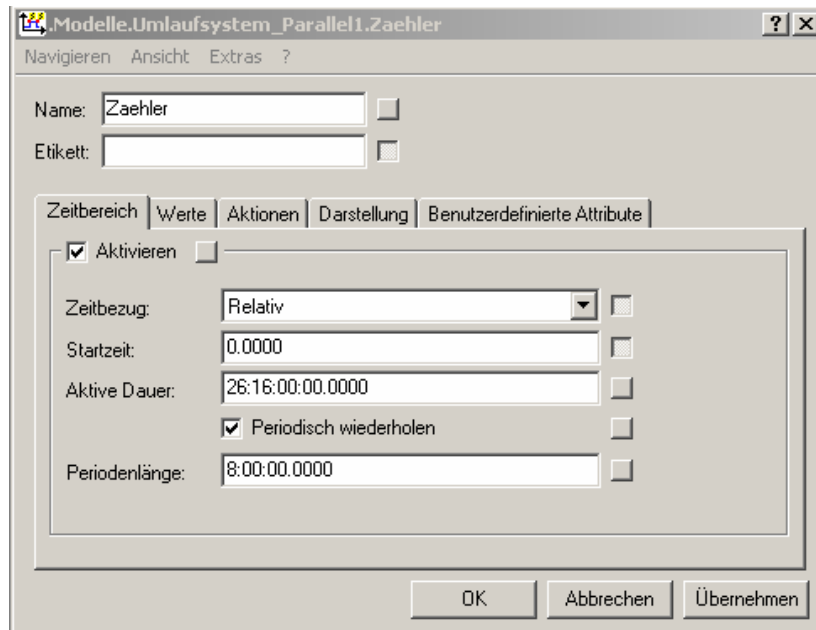


Abbildung 26: Zeitbereich des Triggers

3.10 Die Fertigungssteuerung

Im Netzwerk *Fertigg_Steuerung* sind die den Materialfluss steuernde Tabelle *Arbeitsplan* und die Methode *naechsteStation* enthalten.

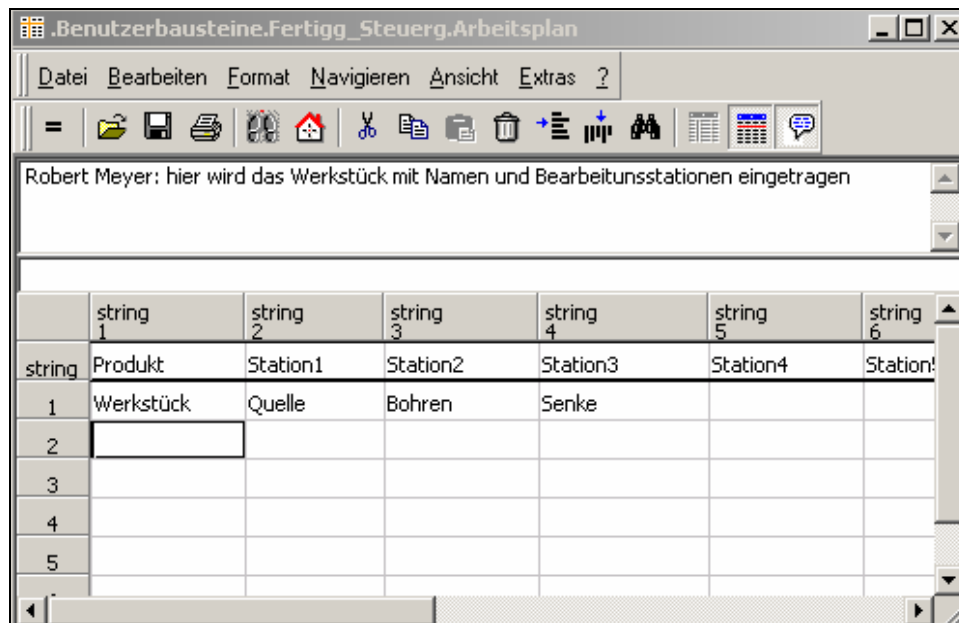
3.10.1. Der Arbeitsplan

Der Arbeitsplan gibt die Bearbeitungsreihenfolge der Arbeitsgänge eines Werkstücks vor. Die Tabelle *Arbeitsplan* enthält in der ersten Spalte die Produktbezeichnung. In unserer Simulation entspricht dies dem in der Klassenbibliothek angelegten BE *Werkstück*. In den folgenden Zeilen können weitere Produkte hinterlegt werden²⁷.

Die weiteren Spalten enthalten die vom Produkt zu durchlaufenden Stationsbezeichnungen. Zur besseren Orientierung sind die Spaltenüberschriften nummeriert. Die hinterlegten Werte müssen der Variable *StationName* des gewünschten Netzwerkes entsprechen. (Beachte Rechtschreibfehler!)

²⁶ über Befehlszeile 28 der Methode Informationsausgabe; vgl. Abbildung 34, Seite 36

²⁷ Diese müssen dann natürlich noch ins System eingespeist werden, ggf. über eine weitere mit dem Eingang des Bausteins *Quelle_WS* verknüpfte Methode



	string 1	string 2	string 3	string 4	string 5	string 6
string	Produkt	Station1	Station2	Station3	Station4	Station!
1	Werkstück	Quelle	Bohren	Senke		
2						
3						
4						
5						

Abbildung 27: Arbeitsplan

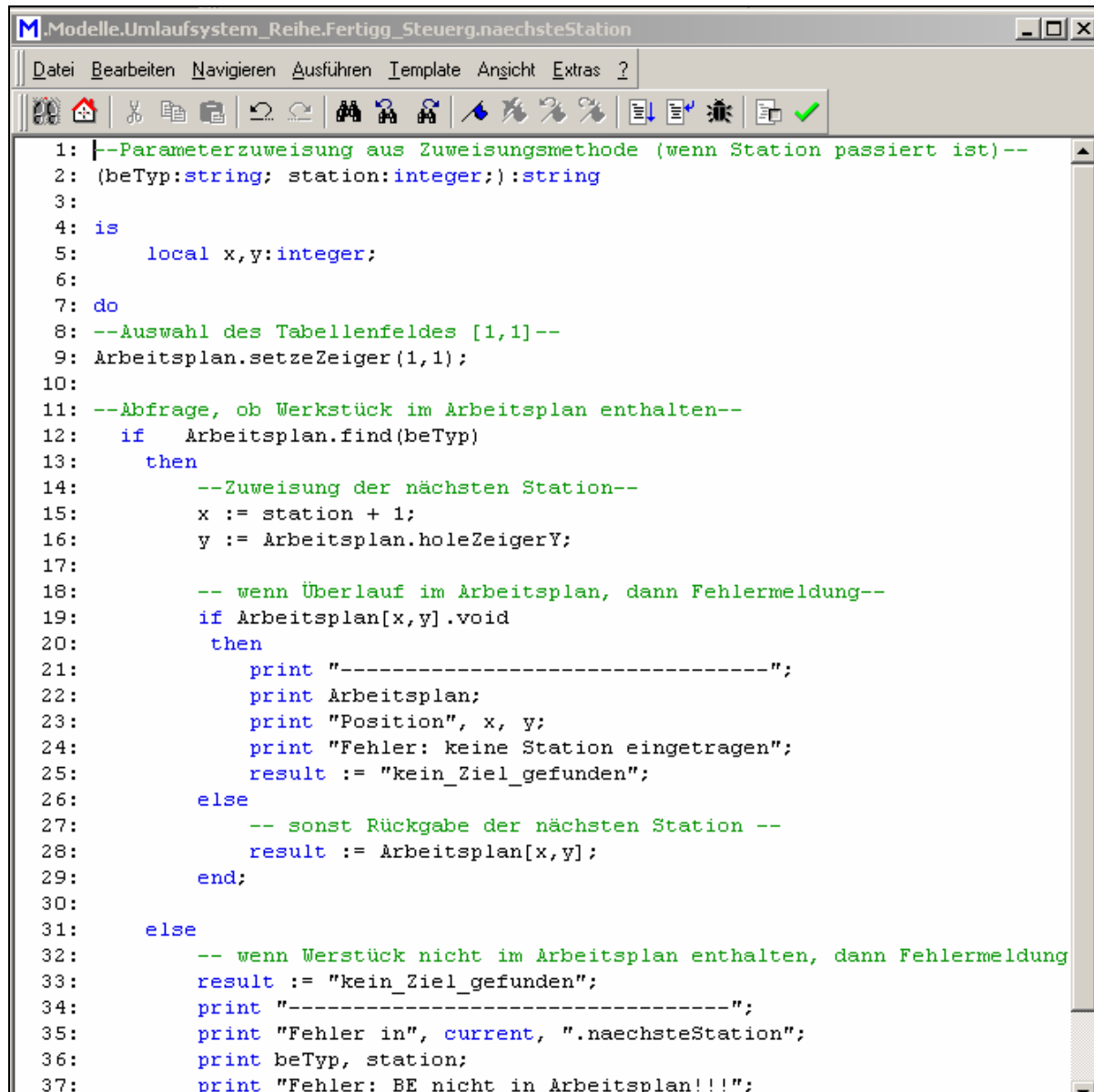
So enthält die Spalte 2 (Station1) der Tabelle immer den Wert *Quelle* - als erste Station im Leben eines Werkstücks. Die Variablen *StationName* der zu durchlaufenden Bearbeitungsnetzwerke werden entsprechend den Simulationsvorgaben in den nachfolgenden Spalten hinterlegt. Die hierbei eingetragene Reihenfolge bestimmt die Bearbeitungsreihenfolge. Eine Station kann ggf. auch mehrmals angefahren werden. Die letzte Station muss eine *Senke* sein, andernfalls kreisen die Werkstückträger immerfort im Umlaufsystem.

3.10.2. Die Methode nächste Station

Die Methode *naechsteStation* ist das Bindeglied zwischen der Tabelle *Arbeitsplan* und den aufrufenden Methoden (*neues*) *Zielzuweisen* (vgl. Abbildung 13) in den Bearbeitungsstationen.

Der Methode *naechsteStation* werden hierbei die Objektbezeichnung „Werkstück“ und das benutzerdefinierte Attribut *BearbeitSchritt* in die Variablen *beTyp* und *station* übergeben.

Die Methode gleicht den Namen des Objektes mit den in Spalte 1 des Arbeitsplanes enthaltenen Produkten ab (siehe Befehlszeile 12 in Abbildung 28).



```
M:\Modelle.Umlaufsystem_Reihe.Fertigg_Steuerg.naechsteStation
Datei Bearbeiten Navigieren Ausführen Template Ansicht Extras ?
1: |--Parameterzuweisung aus Zuweisungsmethode (wenn Station passiert ist)--
2: (beTyp:string; station:integer):string
3:
4: is
5:     local x,y:integer;
6:
7: do
8: --Auswahl des Tabellenfeldes [1,1]--
9: Arbeitsplan.setzeZeiger(1,1);
10:
11: --Abfrage, ob Werkstück im Arbeitsplan enthalten--
12: if Arbeitsplan.find(beTyp)
13:     then
14:         --Zuweisung der nächsten Station--
15:         x := station + 1;
16:         y := Arbeitsplan.holeZeigerY;
17:
18:         -- wenn Überlauf im Arbeitsplan, dann Fehlermeldung--
19:         if Arbeitsplan[x,y].void
20:             then
21:                 print "-----";
22:                 print Arbeitsplan;
23:                 print "Position", x, y;
24:                 print "Fehler: keine Station eingetragen";
25:                 result := "kein_Ziel_gefunden";
26:             else
27:                 -- sonst Rückgabe der nächsten Station --
28:                 result := Arbeitsplan[x,y];
29:             end;
30:
31:     else
32:         -- wenn Werkstück nicht im Arbeitsplan enthalten, dann Fehlermeldung
33:         result := "kein_Ziel_gefunden";
34:         print "-----";
35:         print "Fehler in", current, ".naechsteStation";
36:         print beTyp, station;
37:         print "Fehler: BE nicht in Arbeitsplan!!!";
```

Abbildung 28: Methode naechsteStation

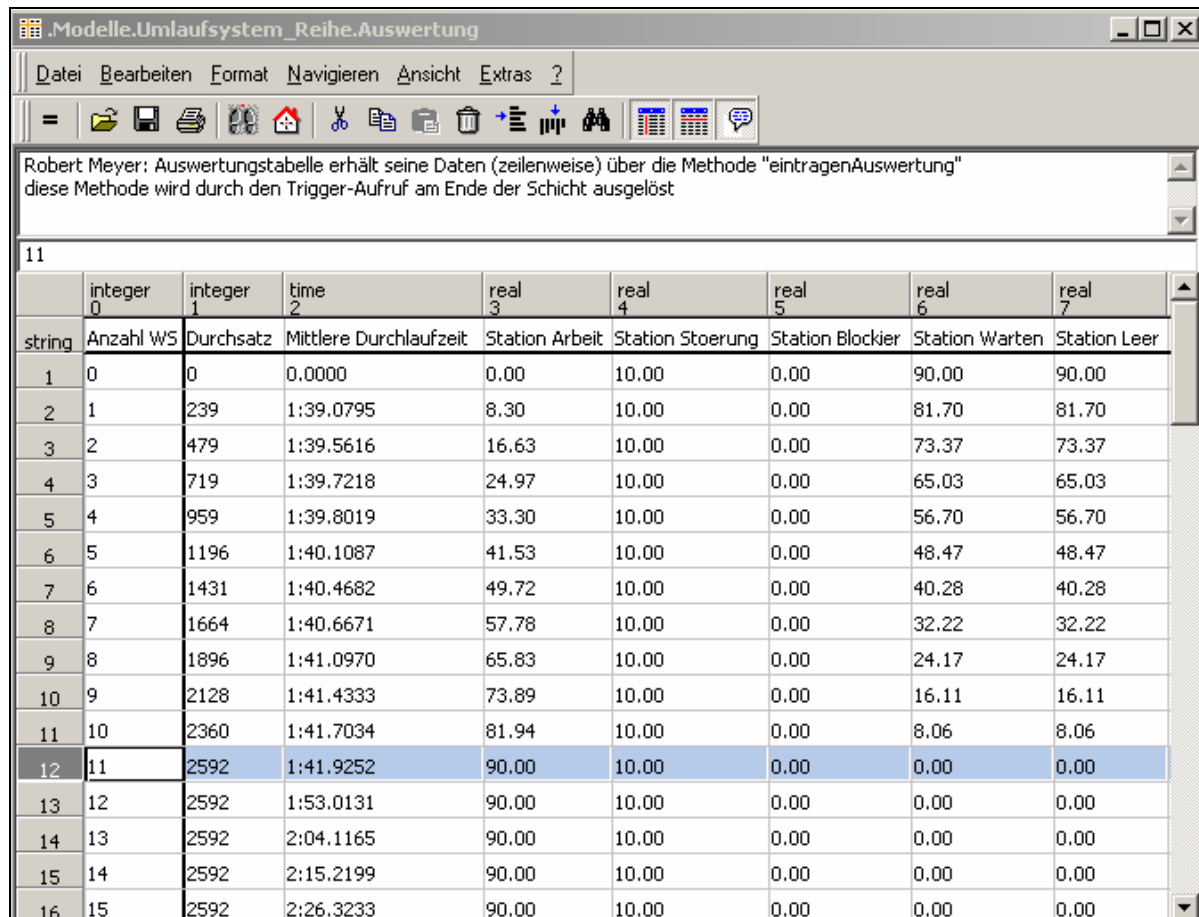
Wurde das Produkt nicht im Arbeitsplan gefunden, erscheint eine Fehlermeldung „*Produkt nicht gefunden!*“ im Direktfenster. Wurde das Produkt im Arbeitsplan gefunden, so sucht die Methode in x-Richtung (spaltenweise) nach der, dem Bearbeitungsschritt nächstfolgenden Arbeitsstation. Existiert eine solche Station – das entsprechende Feld im Arbeitsplan [x,y] ist beschrieben – dann gibt die Methode *naechsteStation* den Inhalt dieses Feldes an die aufrufende Methode zurück. Existiert die nächstfolgende Station nicht (d.h. Arbeitsplatz [x,y] = void), dann erscheint eine Fehlermeldung „*Fehler: keine Station eingetragen*“ im Direktfenster und dem benutzerdefinierten Attribut *Ziel* wird der Eintrag „*kein_Ziel_gefunden*“ zugewiesen.

Sollte eine solche Fehlermeldung erscheinen, ist zuerst der Arbeitsplan nachzuprüfen. Wahrscheinlich wurde der Eintrag der Senke in den Arbeitsplan vergessen.

3.11 Die Tabelle Auswertung

Die Tabelle *Auswertung* enthält die für die Simulation relevanten Ergebnisse.

In den Spaltenüberschriften wird die Bezeichnung der gewünschten Größe eingetragen und der jeweilige Datentyp zugeordnet²⁸, z.B. der Datentyp *time* für die mit Durchlaufzeit überschriebene Spalte 2 in der Tabelle *Auswertung*.



	integer 0	integer 1	time 2	real 3	real 4	real 5	real 6	real 7
string	Anzahl WS	Durchsatz	Mittlere Durchlaufzeit	Station Arbeit	Station Stoerung	Station Blockier	Station Warten	Station Leer
1	0	0	0.0000	0.00	10.00	0.00	90.00	90.00
2	1	239	1:39.0795	8.30	10.00	0.00	81.70	81.70
3	2	479	1:39.5616	16.63	10.00	0.00	73.37	73.37
4	3	719	1:39.7218	24.97	10.00	0.00	65.03	65.03
5	4	959	1:39.8019	33.30	10.00	0.00	56.70	56.70
6	5	1196	1:40.1087	41.53	10.00	0.00	48.47	48.47
7	6	1431	1:40.4682	49.72	10.00	0.00	40.28	40.28
8	7	1664	1:40.6671	57.78	10.00	0.00	32.22	32.22
9	8	1896	1:41.0970	65.83	10.00	0.00	24.17	24.17
10	9	2128	1:41.4333	73.89	10.00	0.00	16.11	16.11
11	10	2360	1:41.7034	81.94	10.00	0.00	8.06	8.06
12	11	2592	1:41.9252	90.00	10.00	0.00	0.00	0.00
13	12	2592	1:53.0131	90.00	10.00	0.00	0.00	0.00
14	13	2592	2:04.1165	90.00	10.00	0.00	0.00	0.00
15	14	2592	2:15.2199	90.00	10.00	0.00	0.00	0.00
16	15	2592	2:26.3233	90.00	10.00	0.00	0.00	0.00

Abbildung 29: Tabelle Auswertung

Die Spaltenanzahl der Tabelle *Auswertung* ist grds. unbegrenzt. Für unseren Zweck reichen die in Abbildung 29 erfassten Daten aus.

Die Ergebnisse werden am Ende jeder Schicht durch die Methode *eintragenAuswertung* in die Tabelle geschrieben²⁹. Am Ende der Simulation sollte die Tabelle *Auswertung* manuell als .txt oder .csv Datei gespeichert werden³⁰, da ein Reset/Neustart der Simulation den Inhalt löscht/überschreibt. Nun kann es für komplexere Simulationen notwendig sein, weitere Daten in der Auswertungstabelle zu erfassen und zu speichern.

²⁸ zum Anlegen von Tabellen vgl. eM-Plant Objects Manual, S. 380; Datentyp der Spalte und Datentyp des einzutragenden Wertes müssen übereinstimmen

²⁹ siehe hierzu Abschnitt 3.13.2

³⁰ diese Datei kann ggf. in MS Excel weiterbearbeitet werden

Dafür sind zunächst die weiteren Spaltenbezeichnungen einzutragen und deren Datentypen festzulegen.

Alsdann ist in der Methode *eintragenAuswertung* eine Zuweisung auf die entsprechende Tabellenspalte einzufügen.

3.12 Variablen

Tecnomatix eM-Plant bietet Variablen zum Speichern von global genutzten Werten an. In der Simulation werden Variablen als vom Bearbeiter einstellbare Parameter sowie als Zählvariablen verwendet.

3.12.1. Untergrenze

Der vom Bearbeiter einstellbare Parameter *Untergrenze* bestimmt die beim Start der Simulation eingeschleuste Anzahl Werkstückträger. Der Parameter ist als Datentyp *integer* definiert und wird vom Bearbeiter im Zusammenwirken mit dem Parameter *Obergrenze* zur Eingrenzung des Untersuchungsbereiches verwendet.

3.12.2. Obergrenze

Der Parameter *Obergrenze* gibt die Anzahl der Werkstückträger vor, welche als Höchstgrenze innerhalb einer Simulationsreihe in das System eingeschleust werden sollen. Ist die *Obergrenze* größer gewählt als die tatsächlich in das Umlaufsystem passende Anzahl Werkstückträger, so blockiert das System die überschüssigen WST. In der Auswertung werden die entsprechenden Werte mit Null eingetragen (das System blockiert).

Ist die *Obergrenze* kleiner als die *Untergrenze* gewählt (z.B. durch Eingabefehler), so erscheint im Direktfenster eine Fehlermeldung, die Simulation bricht ab (Reset).

```
15:     if u>0
16:         then
17:             print"-----";
18:             print"Falsche Eingaben. Bitte korrigieren!";
19:             print"-----";
20:             current.Ereignisverwalter.reset;
21:     end;
```

Abbildung 30: Behandlung Eingabefehler

3.12.3. Maximale Anzahl Werkstückträger (MaxWST)

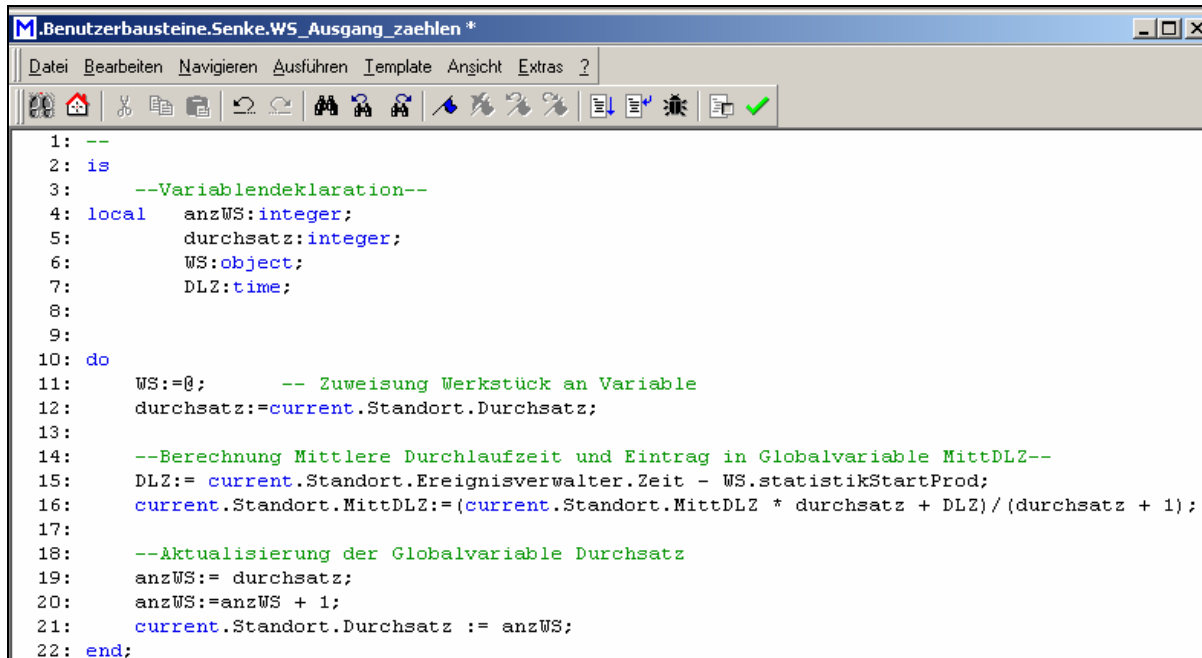
Die Integer-Variable *MaxWST* legt die maximale Anzahl der sich während des **aktuellen** Simulationszyklus (Schicht) im System befindlichen Werkstückträger fest. Die Variable wird in der ersten Schicht durch die Untergrenze belegt. Nach jeder Schicht (also nach 8 Stunden) wird sie um einen Werkstückträger erhöht (siehe Abbildung 3).

3.12.4. Anzahl umlaufender Werkstückträger (AnzUmlaufWST)

Die Integer-Variable *AnzUmlaufWST* erfasst die die sich tatsächlich im System befindlichen Werkstückträger. Sie wird durch jeden neu in das System eingeschleusten Werkstückträger hochgezählt und durch jeden das System über die *Senke_WST* verlassenden Werkstückträger vermindert. Sie stimmt i.d.R. mit der Variable *AnzUmlaufWST* überein, da die Simulation so konzipiert ist, dass die Variable *MaxWST* erreicht wird (siehe Abbildung 11 in Abschnitt 3.5). Nur während eines Hochfahrens des Systems von Null auf die Untergrenze (wenn diese ungleich Null) ergeben sich zeitbedingt Abweichungen zwischen *AnzUmlaufWST* und *MaxWST*.

3.12.5. Mittlere Durchlaufzeit (MittIDLZ)

Die Variable *MittDLZ* ist vom Datentype *time* und wird durch Mittelwertbildung der Durchlaufzeiten der das Umlaufsystem durchlaufenen **Werkstücke** bestimmt. Die Durchlaufzeit (DLZ) eines Werkstückes bemisst sich dabei aus der Differenz von aktueller Zeit (*current.Standort.Ereignisverwalter.Zeit*) bei Aufruf der mit dem Eingang des Bausteins *WS_Senke* verbundenen Methode *WS_Ausgang_zaehlen* und der als Statistikeigenschaft des Werkstücks hinterlegten Startzeit (*WS.statistikStartProd*). Abbildung 31 zeigt dies in den Befehlszeilen 15 und 16:



```
1: --
2: is
3:   --Variablendeklaration--
4: local   anzWS:integer;
5:         durchsatz:integer;
6:         WS:object;
7:         DLZ:time;
8:
9:
10: do
11:   WS:=@;      -- Zuweisung Werkstück an Variable
12:   durchsatz:=current.Standort.Durchsatz;
13:
14:   --Berechnung Mittlere Durchlaufzeit und Eintrag in Globalvariable MittDLZ--
15:   DLZ:= current.Standort.Ereignisverwalter.Zeit - WS.statistikStartProd;
16:   current.Standort.MittDLZ:=(current.Standort.MittDLZ * durchsatz + DLZ) / (durchsatz + 1);
17:
18:   --Aktualisierung der Globalvariable Durchsatz
19:   anzWS:= durchsatz;
20:   anzWS:=anzWS + 1;
21:   current.Standort.Durchsatz := anzWS;
22: end;
```

Abbildung 31: Methode zur Berechnung der mittleren Durchlaufzeit und Durchsatz

Hierbei ist anzumerken, dass die Zeitstatistik des Werkstücks bei Aufbringen auf einen Werkstückträger in der Methode *WSaufWSTbringen* im Netzwerk *Quelle* initialisiert wird. Daraus

folgt, dass sich der Startzeitpunkt nicht auf das Bereitlegen an der Quelle³¹, sondern auf das Aufbringen auf den Werkstückträger bezieht. Durch Löschen der Befehlszeile 8 in der Methode *WSaufWSTbringen* (vgl. Abbildung 12 im Abschnitt 3.5) kann der Startzeitpunkt ggf. auf den Zeitpunkt des Bereitlegens an der Quelle zurückgesetzt werden.

3.12.6. Durchsatz

Die Integervariable *Durchsatz* erfasst alle die Senke innerhalb einer Schicht erreichenden Werkstücke. Sie wird bei jedem in die *Senke_WS* des Netzwerkes *Senke* eingehenden Werkstück durch die Methode *WS_Ausgang_zahlen* um eins erhöht (siehe Abbildung 31, Zeilen 18 bis 21).

Der *Durchsatz* wird am Ende jeder Schicht in die Auswertungstabelle eingetragen und anschließend auf Null gesetzt.

3.12.7. Bandgeschwindigkeit

Der Parameter *Bandgeschwindigkeit* ist vom Datentyp *speed* und legt die Umlaufgeschwindigkeit des Bandes, der Ecken und sonstigen Einzelstationen fest, soweit dort keine Bearbeitung erfolgt. Der Parameter *Bandgeschwindigkeit* wirkt über die Methode *Parameter* (vgl. Abschnitt 3.13.4, Seite 37) auf o.g. Bausteine. Hierdurch ist eine einfache, konsistente und zentrale Einstellung der Bandgeschwindigkeit durch den Bearbeiter möglich.

3.12.8. Bearbeitungszeit

Der Parameter *Bearbeitungszeit* ist vom Datentyp *time* und geht über entsprechende Methoden (z.B. *Bearbeitung* in den Netzwerken *Reihe-/ Ausschleusstation*) in die Bearbeitungszeit der Bearbeitungsstationen ein. In unseren Versuchen wurde der Parameter *Bearbeitungszeit* auf 10 s festgelegt (siehe Tabelle 2). Durch die globale Festlegung ist ein einheitlicher Takt der Bearbeitungsstationen gewährt. Selbstverständlich kann die Bearbeitungszeit im Bedarfsfall abweichend auch in jeder Einzelstation festgelegt werden³².

In einer perspektivischen Weiterentwicklung des Modells ist eine werkstück- sowie stationsabhängige Festlegung der Bearbeitungszeit im Arbeitsplan denkbar.

³¹ zu diesem Zeitpunkt setzt das Programm den Wert *statistikStartProd* in der Statistiktabelle des Werkstücks.

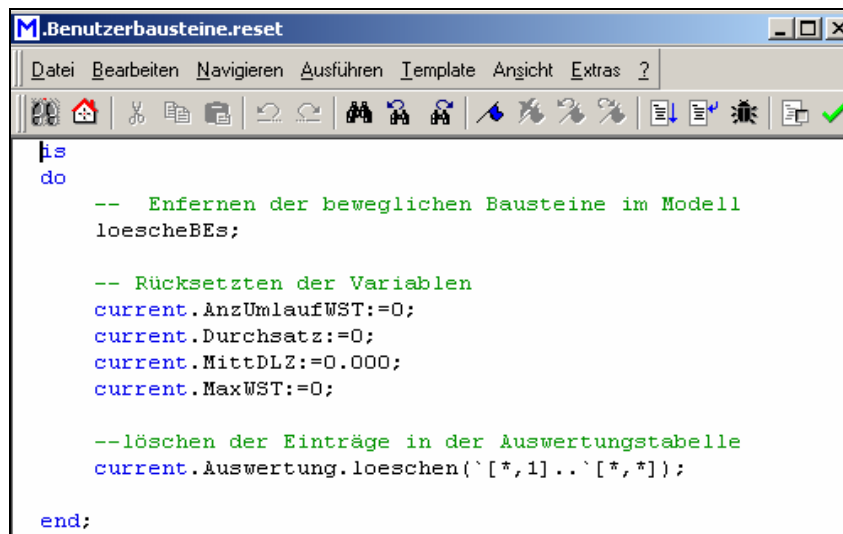
³² dies muss dann in der jeweiligen Methode erfolgen; ein manueller Eintrag in der Registerkarte im Feld Bearbeitungszeit der Station würde von der Methode *Bearbeitung* überschrieben werden.

3.13 Methoden auf Netzwerkebene

Neben den Methoden in den einzelnen Netzwerken enthält das Grundmodell Umlaufsystem Methoden zur Steuerung der Objekte sowie zur Auswertung der Ergebnisse auf oberster Netzwerkebene.

3.13.1. Reset

Die Methode *Reset* ist eng mit dem Ereignisverwalter verbunden. Sie wird beim Betätigen des Buttons Reset im Ereignisverwalter aufgerufen. Dies sollte vor jedem Neustart der Simulation geschehen.



```
is
do
  -- Entfernen der beweglichen Bausteine im Modell
  loescheBEs;

  -- Rücksetzten der Variablen
  current.AnzUmlaufWST:=0;
  current.Durchsatz:=0;
  current.MittDLZ:=0.000;
  current.MaxWST:=0;

  --löschen der Einträge in der Auswertungstabelle
  current.Auswertung.loeschen(`[*],1)..`[*],*`);

end;
```

Abbildung 32: Methode Reset

Wie in Abbildung 32 ersichtlich, werden durch die Reset-Methode alle Werkstücke und Werkstückträger aus dem Umlaufsystem entfernt, die aufgeführten Zählvariablen zurückgesetzt und die Einträge in der Auswertungstabelle gelöscht.

3.13.2. Die Methode eintragenAuswertung

Über die Methode *eintragenAuswertung* werden nach jedem Schichtende (also alle 8 h) die Statistikdaten entsprechend den Befehlszeilen 25 bis 33 in Abbildung 33 in die Auswertungstabelle eingetragen.


```

Datei Bearbeiten Navigieren Ausführen Template Ansicht Extras ?
[Icons]
2:      --Variablenübergabe vom Trigger--
3: (a,b :boolean)
4: is
5:      -- Variablendeklaration--
6:  tabelle:object;
7:  station:object;
8:  MittDLZ:time;
9:  durchsatz,o,u,y:integer;
10: do
11:  --Zuweisung an Variablen--
12:  tabelle:= current.Auswertung;
13:  durchsatz:= current.Durchsatz;
14:  y:= current.MaxWST+1;           -- gibt Zeile vor
15:  o:= current.Obergrenze;        -- gibt festgelegte Obergrenze wieder
16:  u:= current.Untergrenze;      -- gibt festgelegte Untergrenze wieder
17:  --***WICHTIG!!!***--
18:  --***An dieser Stelle Bezeichnung der zu analysierenden Station(en) eintragen!***
19:  station:=current.Parallel_Bearbeitung.Bearbeitung;
20:  MittDLZ:= current.MittDLZ; -- weist Senke als Objektvariable zu
21:  -- diese Bedingung führt beim Start der Simulation zum Überspringen des Eintrages
22:  if current.Ereignisverwalter.Zeit > 0.000
23:  Then
24:    --Eintragen der Statistikwerte in die Auswertungstabelle--
25:    tabelle[0,y-u] := y-1;      -- (y-u) gibt die Zeile in der Statistik vor
26:    tabelle[1,y-u] := durchsatz;
27:    tabelle[2,y-u] := MittDLZ;
28:    tabelle[3,y-u] := station.statArbeitsAnteil * 100;
29:    tabelle[4,y-u] := station.statStoerungsAnteil *100;
30:    tabelle[5,y-u] := station.statBlockiertAnteil*100;
31:    tabelle[6,y-u] := station.statWarteAnteil*100;
32:    tabelle[7,y-u] := station.statLeerAnteil*100;
33:    tabelle[8,y-u] := station.statPausenAnteil*100;
34:    --hier können beliebige weitere Werte eingetragen werden--
35:    --(Spaltenformat der Auswertungstabelle beachten)--
36:    current.MaxWST := y;      -- zählt weiter (MaxWST ist zugleich Zählvariable)
37:    current.Durchsatz := 0; -- Rücksetzen des Durchsatzes nach Schichtende
38:  else
39:    -- Gesamtdauer aus Periodenlänge übernehmen --
40:    current.Zaehler.aktiveDauer := current.Zaehler.periodenLaenge * (o-u) ;
41:    -- Setzen der Start-Werkstückträgeranzahl--
42:    current.MaxWST:=u;
43:  end;
44:  -- Rücksetzen der Statistik am Periodenende--
45:  current.Senke.Senke_WS.initStat;

```

Abbildung 33: Methode eintragenAuswertung

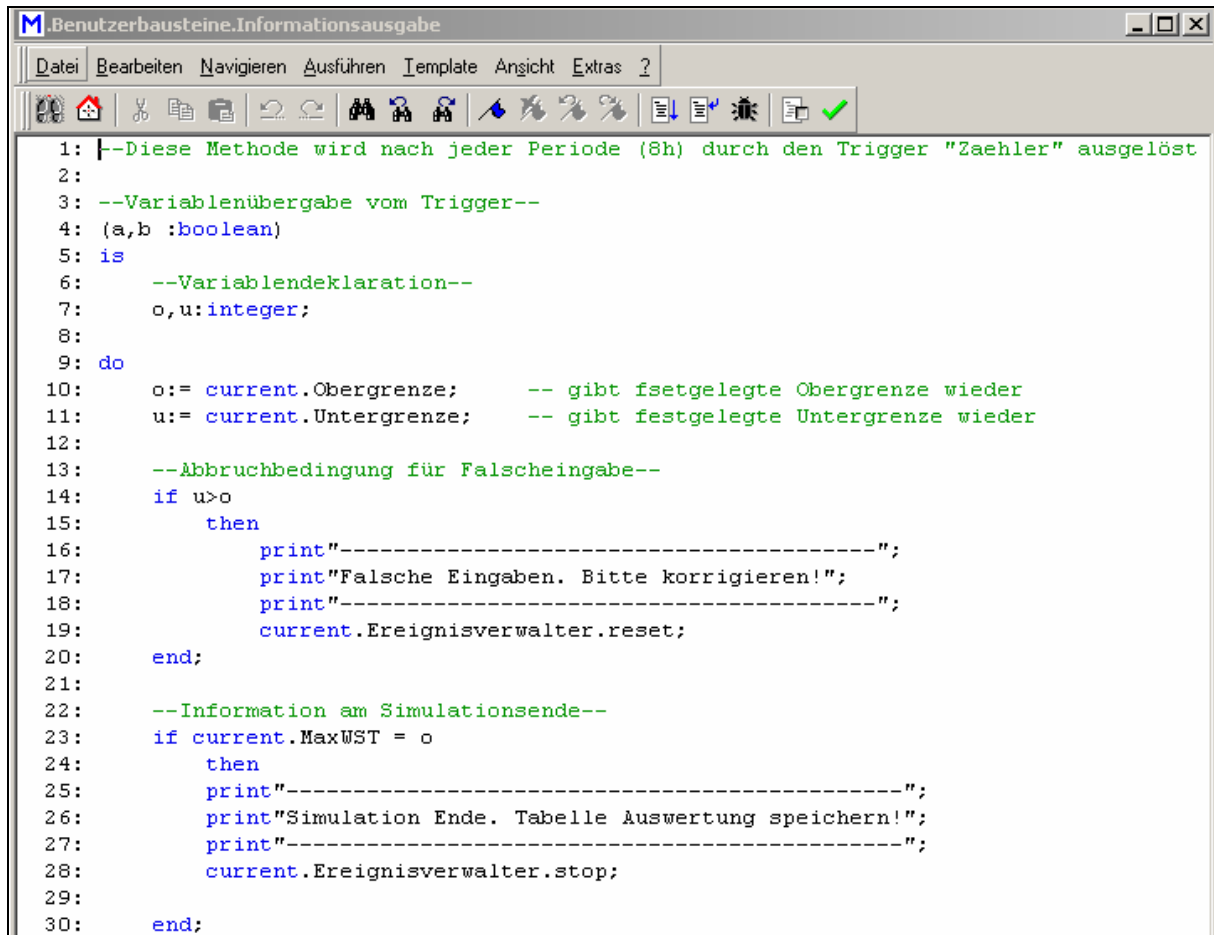
Dabei wird die jeweilige Tabellenzeile aus der Differenz der Variablen y und u gebildet. Diese bekommen ihre Werte von den Globalvariablen $MaxWST$ (y) und $Untergrenze$ (u) übergeben. Da sich die Variable $MaxWST$ bei jedem Schichtende um eins erhöht (vgl. Zeile 14 und Zeile 36 in Abbildung 33), werden die Daten jeder Schicht in eine neue Zeile eingetragen. Beim Start der Simulation (Zeitpunkt 0.00) legt die Methode die Dauer der Simulation im Trigger $Zaehler$ fest, und übergibt der Variablen $MaxWST$ der Wert der Untergrenze u (vgl. Zeilen 39 bis 42 in Abbildung 33).

Die Methode *eintragenAuswertung* bewirkt letztlich das Zurücksetzen der Statistik im Baustein *Senke_WS* am Ende jeder Schicht. Dies ist notwendig, weil die Variable *Durchsatz* auf diese Statistik zurückgreift³³.

3.13.3. Die Methode Informationsausgabe

Die vom Trigger *Zaehler* ausgelöste Methode *Informationsausgabe* hat zwei Aufgaben:

1. Im Fall einer Falscheingabe von Ober- und Untergrenze erscheint eine Fehlermeldung im Direktfenster und die Simulation bricht ab.
2. Am Ende der Simulation (MaxWST = Obergrenze) wird der Ereignisverwalter gestoppt und im Direktfenster eine Aufforderung zum Speichern der Auswertung ausgegeben.



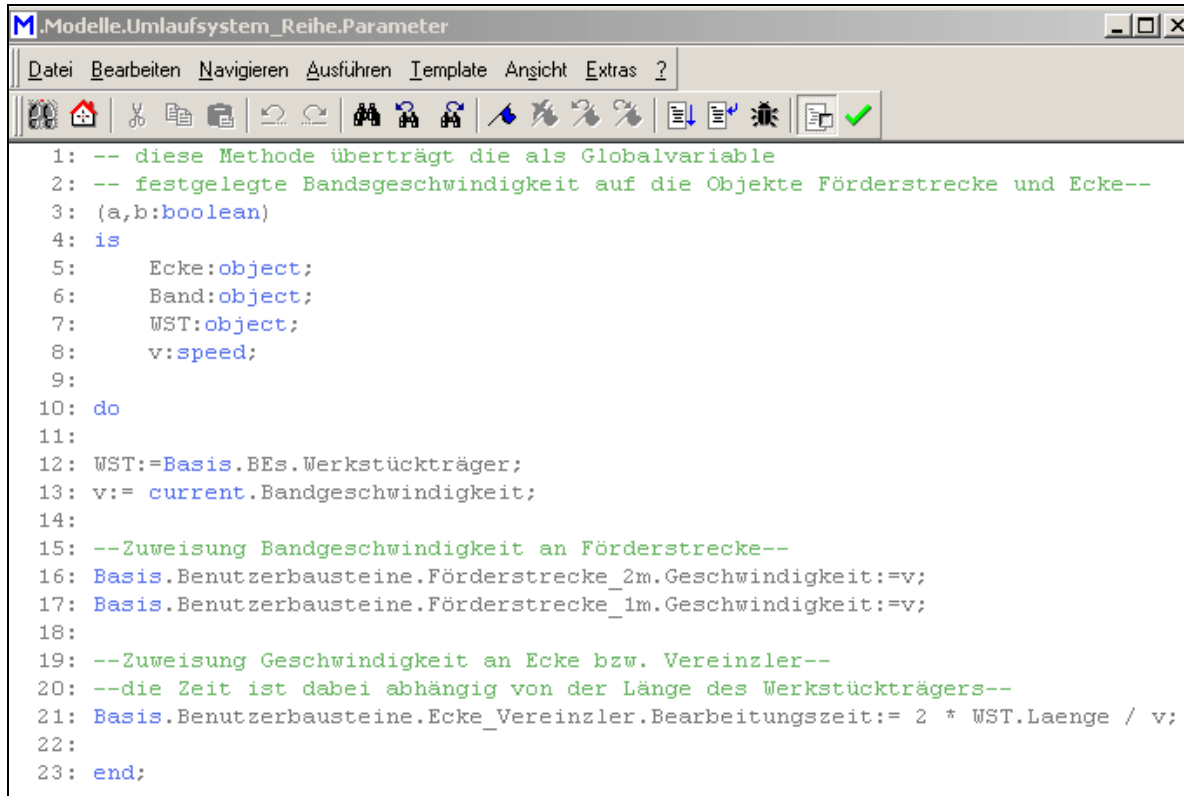
```
1: |--Diese Methode wird nach jeder Periode (8h) durch den Trigger "Zaehler" ausgelöst
2:
3: --Variablenübergabe vom Trigger--
4: (a,b :boolean)
5: is
6:   --Variablendeklaration--
7:   o,u:integer;
8:
9: do
10:  o:= current.Obergrenze;    -- gibt fsetgelegte Obergrenze wieder
11:  u:= current.Untergrenze;   -- gibt festgelegte Untergrenze wieder
12:
13:  --Abbruchbedingung für Falscheingabe--
14:  if u>o
15:    then
16:      print"-----";
17:      print"Falsche Eingaben. Bitte korrigieren!";
18:      print"-----";
19:      current.Ereignisverwalter.reset;
20:    end;
21:
22:  --Information am Simulationende--
23:  if current.MaxWST = o
24:    then
25:      print"-----";
26:      print"Simulation Ende. Tabelle Auswertung speichern!";
27:      print"-----";
28:      current.Ereignisverwalter.stop;
29:
30:  end;
```

Abbildung 34: Methode *Informationsausgabe*

³³ vgl. Abschnitt 3.12.6 und Abbildung 31

3.13.4. Die Methode Parameter

Die ebenfalls mit dem Trigger verknüpfte Methode *Parameter* setzt die vom Bearbeiter eingestellte Variable *Bandgeschwindigkeit* an die Objekte.



```
1: -- diese Methode überträgt die als Globalvariable
2: -- festgelegte Bandgeschwindigkeit auf die Objekte Förderstrecke und Ecke--
3: (a,b:boolean)
4: is
5:   Ecke:object;
6:   Band:object;
7:   WST:object;
8:   v:speed;
9:
10: do
11:
12: WST:=Basis.BEs.Werkstückträger;
13: v:= current.Bandgeschwindigkeit;
14:
15: --Zuweisung Bandgeschwindigkeit an Förderstrecke--
16: Basis.Benutzerbausteine.Förderstrecke_2m.Geschwindigkeit:=v;
17: Basis.Benutzerbausteine.Förderstrecke_1m.Geschwindigkeit:=v;
18:
19: --Zuweisung Geschwindigkeit an Ecke bzw. Vereinzler--
20: --die Zeit ist dabei abhängig von der Länge des Werkstückträgers--
21: Basis.Benutzerbausteine.Ecke_Vereinzler.Bearbeitungszeit:= 2 * WST.Laenge / v;
22:
23: end;
```

Abbildung 35: Die Methode Parameter

Den Bausteinen *Förderstrecke_1m* bzw. *Förderstrecke_2m* wird der Wert der Variable *Bandgeschwindigkeit* auf die Objekteigenschaft *Geschwindigkeit* direkt zugewiesen, da beide vom gleichen Datentyp sind.

Der Objekttyp *Ecke_Vereinzler* hat keine solche Eigenschaft vom Datentyp *speed*. Deshalb muss der Datentyp *speed* in den Datentyp *time* umgerechnet werden.

Aus der bekannten Formel $v = \frac{s}{t}$ folgt $t = \frac{s}{v}$.

Der Weg s entspricht dabei der zweifachen Werkstückträgerlänge³⁴, die Geschwindigkeit v entspricht der Variable *Bandgeschwindigkeit*.

So berechnet die Methode *Parameter* aus der Division der zweifachen Werkstückträgerlänge³⁵ durch die Bandgeschwindigkeit die entsprechend benötigte Zeit im Objekttyp *Ecke_Vereinzler* und weist sie diesem als Bearbeitungszeit zu.

³⁴ unter der Annahme, dass der Werkstückträger quadratisch ist

³⁵ die Werkstückträgerlänge ist als Eigenschaft im Klassenobjekt *Werkstückträger* einstellbar

4. Ergebnisse der Modelle

4.1 Plausibilitätstest

Um die aufgebaute Simulation zu validieren, wurden verschiedene Plausibilitätsüberlegungen angestellt. Dazu wurden die drei Umlaufsysteme

1. Bearbeitung in Reihe³⁶
2. Ausschleusstation Variante 1
3. Ausschleusstation Variante 2

simuliert: Je einmal ohne Störung und je einmal mit zeitlich 10%iger Störung an der Bearbeitungsstation.

Hierbei wird auf die in Tabelle 2 (siehe Seite 2) dargestellten Grundparameter zurückgegriffen. Das Ergebnis der Simulation ist in den im Anhang aufgeführten Tabelle 3 bis Tabelle 8 dokumentiert.

4.1.1. Überlauf

Einfach ist der Test bei Überlauf des Systems mit Werkstückträgern. Es können nur so viele Werkstückträger in das System eingebracht werden, wie Kapazität vorhanden ist. Die Kapazität wurde auf drei Werkstückträger (0,3m Länge) je 1 m Förderstrecke festgelegt. Eine Station bzw. Ecke kann jeweils einen Werkstückträger aufnehmen. Beim Überladen (Anzahl Werkstückträger > Kapazität) wird die Simulation blockiert. In die Auswertungstabelle wird „Null“ eingetragen bis die Obergrenze erreicht ist.

4.1.2. Engpass

Es ist in den Simulationen leicht erkennbar, dass die Bearbeitungsstation mit 10s Bearbeitungszeit den Engpass darstellt. Optisch sieht man dies an der Staubildung bei Erreichen der 100%igen Auslastung des Bausteins *Bearbeitung*.

Gleiches lässt sich aus der Programmlogik ableiten:

Die Verweilzeit des Werkstückträgers in Quelle und Senke beträgt 4 s.

Die Verweilzeit des Werkstückträgers in den Ecken beträgt 3 s.

Die Zeit für einen entsprechenden Abschnitt (0,3m) auf der Förderstrecke beträgt 1,5 s bei einer Bandgeschwindigkeit 0,2 m/s.

³⁶ Abbildung 2 zeigt das Umlaufsystem *Bearbeitung_Reihe*

Es bereits aus der Montagetechnik bekannt, dass zur Optimierung eines Systems immer der Engpass betrachtet werden muss und im Idealfall eine Angleichung der Taktzeiten und Kapazitäten erfolgen soll.³⁷

4.1.3. Optimale Werkstückträgeranzahl

Die Optimale Werkstückträgeranzahl in einem Umlaufsystem ist dann erreicht, wenn der Durchsatz unter den gegebenen Parametern und Berücksichtigung weiterer Größen wie der Durchlaufzeit und der Kostenfunktion³⁸ maximal wird. Wie aus Abbildung 36 und Abbildung 37 erkennbar, ist dies bei Erreichen der Kapazitätsgrenze am Engpass Bearbeitungsstation der Fall. Unter den gegebenen Bedingungen liegt die optimale die Anzahl Werkstückträger je nach Länge des Umlaufsystems bei 11 bzw. 12 Stück (vgl. Tabellen 3 bis 8).

4.1.4. Durchsatz

Der Durchsatz gibt die Zahl der erfolgreich in das System eingeschleusten, bearbeiteten und in der Senke ausgeschleusten Werkstücke während einer Schicht an. Wie in Abbildung 36 erkennbar, steigt der Durchsatz in allen Versuchsreihen linear bis zum Erreichen der 100%igen Auslastung der Engpasstation (entspricht optimaler Werkstückträgeranzahl) an.

Es leicht erkennbar, dass der Engpass, die Bearbeitungsstation mit 10 s Bearbeitungszeit je Werkstück, den Durchsatz determiniert.

Man teile die Dauer der Schicht, also 8 h (28.800 s) durch 10 s (Bearbeitungszeit/Teil).

Ergebnis: 2.880 Teile

Dies entspricht genau dem in den Tabellen 3, 5 und 7 erfassten Durchsatz bei 100 %iger Auslastung der Engpasstation. Eine Erhöhung der Bandgeschwindigkeit bzw. Verkürzung der Zeit in Quelle und Senke (aktuell je 4s) würde den Durchsatz **nicht** erhöhen.

³⁷ die Stationen mit geringerem Zeitbedarf sind nicht ausgelastet bzw. durch Stau blockiert

³⁸ wird hier nicht weiter betrachtet, es ist jedoch anzunehmen, dass jeder zusätzliche Werkstückträger Kosten verursacht (Anschaffung, Kapitalkosten, Wartung, Lager)

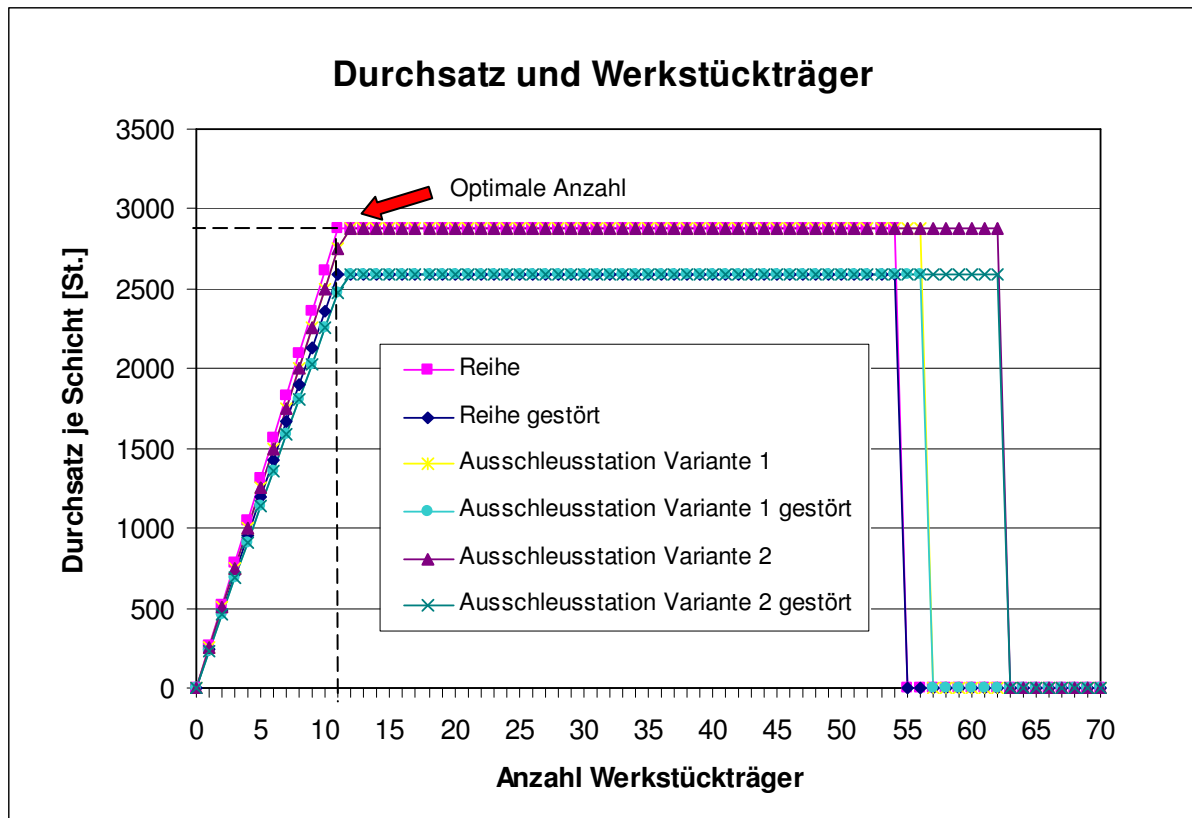


Abbildung 36: Abhängigkeit von Durchsatz und Anzahl der Werkstückträger (8h-Schicht)

Es ist logisch, dass der Durchsatz von 2.880 Werkstücken je Schicht (ungestört), bzw. 2.592 Werkstücken (bei 10%iger Störung) in allen Modellen gleich ist, da der Durchsatz durch den Engpass bestimmt wird.

Je nach Länge des Umlaufsystems (Länge der Förderstrecke, Anzahl Ecken bzw. Ausschleusstationen) wird bei einer gewissen Anzahl Werkstückträger die Kapazitätsgrenze des Umlaufsystems erreicht und dieses blockiert. Folglich fällt der Durchsatz auf Null zurück. Die Netzwerke *Ausschleusung* unterscheiden sich von der Bearbeitungsstation in Reihe durch zwei zusätzliche Stationen zur Aus- bzw. Einschleusung. Das Umlaufsystem *Ausschleusstation Variante 2* kann durch Auslastung der Parallelstrecke (2 Meter) sechs zusätzliche Werkstückträger aufnehmen.

4.1.5. Störung

Gemäß der Anforderung von Prof. Jacobs soll die Simulation zusätzlich mit einer Störung an der Bearbeitungsstation getestet werden. eM-Plant bietet zur Störungsabbildung vielfältige Möglichkeiten von zyklischer Störung bis zu Exponentialverteilung. In unserem Fall wurde eine zyklische Störung von 6 min je Stunde (10%) an der Engpasstation festgelegt. Die Störung schlägt 1:1 auf den Durchsatz – dieser fällt um 10% - durch. Die Durchlaufzeit erhöht sich wegen des Rückstaus an der gestörten Station.

Darüber hinaus wurde eine Störung von 12 min je 2 Stunden, also ebenfalls 10%, an der Engpassstation generiert. In den unseren einfachen Modellen hat die unterschiedliche Verteilung einer vom Umfang gleichen Störung keine Auswirkung. Unter diesem Aspekt interessant scheint jedoch die Koppelung zeitlich abweichend gestörter Stationen.

4.1.6. Durchlaufzeit

Die mittlere Durchlaufzeit wird durch Mittelwertbildung berechnet³⁹. Bis zum Erreichen der 100%en Auslastung der Engpassstation (optimaler Bereich) bleibt sie konstant und entspricht der Summe der Bearbeitungszeiten und den Laufzeit auf den Förderstrecken.

Sind die Zeiten der Simulation plausibel?

Die Richtigkeit zeigt nachfolgende Rechnung am Beispiel der Bearbeitungsstationen

a) in Reihe:

Länge Förderstrecke Quelle bis Senke ⁴⁰ :	12 m
Bandgeschwindigkeit:	0,2 m/s
Anzahl Vereinzler (Ecken) auf dieser Strecke:	4 Stück
Bearbeitungszeit in Quelle und Senke:	je 4 s

Daraus ergibt sich die

$$\textit{TheoretischeBearbeitungszeit} = \frac{12m}{0,2m/s} + 4 \cdot \frac{0,6m}{0,2m/s} + 2 \cdot 4s + 10s = 90s = 1 : 30 \text{ min}$$

b) Parallelstation:

Länge Förderstrecke Quelle bis Senke:	12 m
Bandgeschwindigkeit:	0,2 m/s
Anzahl Vereinzler (Ecken) auf dieser Strecke:	6 Stück
Bearbeitungszeit in Quelle und Senke:	je 4 s

Daraus ergibt sich die

$$\textit{TheoretischeBearbeitungszeit} = \frac{12m}{0,2m/s} + 6 \cdot \frac{0,6m}{0,2m/s} + 2 \cdot 4s + 10s = 96s = 1 : 36 \text{ min}$$

Ergebnis:

Diese theoretischen Durchlaufzeiten stimmen mit den in den Auswertungstabellen erfassten Daten⁴¹ überein.

³⁹ Vgl. Abschnitt 3.12.5

⁴⁰ die Gesamtstrecke des Umlaufsystems beträgt 16 m

Nach Erreichen der optimalen Werkstückträgeranzahl (hier 11 WST) erhöht sich die mittlere Durchlaufzeit linear in allen Versuchsreihen um 10s je Schicht. Logisch?

Ja, denn der Stau vor der Engpassstation wächst mit jedem weiteren suboptimalen Werkstückträger um dessen Bearbeitungszeit von 10 s.

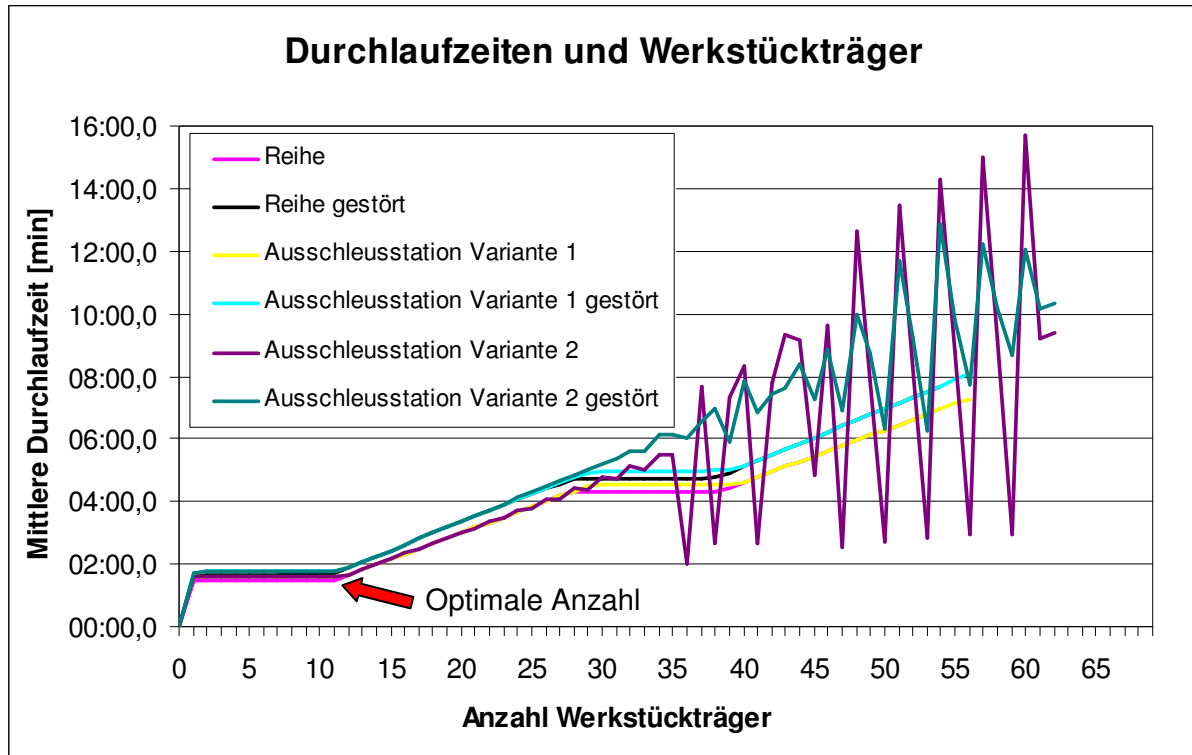


Abbildung 37: Abhängigkeit von Durchlaufzeit und Werkstückträgeranzahl

Bei einer Anzahl von ca. 40 Werkstückträgern im System bleibt die mittlere Durchlaufzeit hingegen konstant. Warum? Nun ebenfalls logisch: Das Ende des Rückstaus der Werkstückträger vom Engpass befindet sich nun zwischen Quelle und Senke. Hier fahren die Werkstückträger jedoch leer. Da für die Berechnung der Durchlaufzeit auf die Werkstücke abgestellt wird, bleiben die sich in diesem Bereich stauenden leeren Werkstückträger unberücksichtigt.

Eine Ausnahme bildet das Modell *Ausschleusstation_Variante_2*:

Hier stauen sich die Werkstückträger nicht an der Engpassstation, sondern fahren im Umlaufsystem weiter. Da die nicht bearbeiteten Werkstücke jedoch noch bearbeitet werden müssen, drehen sie eine (oder auch mehrere) sog. „Ehrenrunden“. Hierbei tritt nun ein interessantes Phänomen auf:

Während einer Schicht gelangen einige Werkstückträger wiederholt nicht in die Ausschleusstation, da der Puffer zum betreffenden Zeitpunkt voll ist. Neu eingeschleuste Werkstücke

⁴¹ selbstverständlich nur bei den ungestörten Stationen bis zum Erreichen der 100 %en Auslastung

hingegen werden sofort bearbeitet. Folge sind extrem geringe Durchlaufzeiten, die fast das Ausgangsniveau (1:36 min) erreichen.

Kommt nun in einer nachfolgenden Schicht ein zusätzlicher Werkstückträger in das System, kehrt sich das Verhalten um: es werden vermehrt „alte“ Werkstücke in die Bearbeitungsstation ausgeschleust. Die mittlere Durchlaufzeit der bearbeitenden Werkstücke steigt extrem an. Ein solcher Zyklus erstreckt sich über fünf Schichten.

Legt man an die betreffenden Kurven jedoch eine Trendfunktion an, so bestätigt sich die gleiche Tendenz wie in den Umlaufsystemen mit Stau am Engpass.

4.1.7. Ergebnis der Plausibilitätstests

Die von der Simulation in den Grundmodellen berechneten Ergebnisse sind plausibel. Es treten keine Syntax und Laufzeitfehler auf. Logische Fehler sind im Hinblick auf die Tests nicht erkennbar.

Damit ist die Simulation hinreichend validiert.

5. Kritik und Ausblick

Abschließend einige kritische Worte:

Mit dem Modell liegt eine Grundlage für die Simulation von Materialflusssystemen analog zur Stein-Anlage vor.

Bearbeitern ist dringend empfohlen, die geforderten Prozesse klar zu definieren und zu strukturieren, bevor mit der eigentlichen Programmierung angefangen wird.

Im Hinblick auf die Simulationssoftware Plant-Simulation sollte es einen Ansprechpartner im Fachbereich geben der sich hinreichend mit der Software auskennt.

Hilfreich und lobenswert ist die Ausgabe von Tutorialmaterial durch Herrn Gruhn.

In diesem Zusammenhang sollte verstärkt Wert auf die Auskommentierung von Programmcode durch die Bearbeiter vorhergehender Projekte gelegt werden.

5.1 Ansätze und Aufbaumöglichkeiten

Mit dem jetzigen Systemstand sind beliebig viele Arbeitsstationen der drei Grundtypen integrierbar; die wesentlichen Simulationsparameter sind zentral einstellbar. Die Bearbeitungsreihenfolge und damit die Materialflussrichtung der Werkstücke werden im Arbeitsplan festgelegt.

Wo kann neue Programmierung ansetzen?

Insbesondere in der Testphase sind Erweiterungsmöglichkeiten aufgefallen, die Änderungen im Methodencode und zusätzlicher Objekte bedürfen.

Denkbar sind:

1. Erhöhung der Werkstückvielfalt (Produktionsprogramm),
2. (unterschiedliche) Bearbeitungszeiten in Abhängigkeit Werkstück/Arbeitsplatz/Arbeitsgang im Arbeitsplan,
3. Visualisierung mit dem 3-D Viewer (erfordert Umwandlung der Netzwerkstruktur),
4. Vergleich von verketteten Ausschleusarbeitsplätzen mit und ohne Entstauungsfunktion.

Es ist wünschenswert, wenn die Simulation in Folgeprojekten erweitert, verbessert und die Funktionalität erhöht wird.

In diesem Sinne

Panta Rhei!

Literaturverzeichnis

Kolb, Katja, Erstellung eines Tutorials für das Simulationstool eM-Plant 7.0,

FB Wirtschaftsingenieurwesen FH Jena, 2005, ausliegend bei Hr. Gruhn

RRZN, Grundlagen der objektorientierten Programmierung, 2. Aufl., 2000

Schulze, Simulation von Materialflusssystemen, Landsberg, 1988

UGS Tecnomatix (Hrsg.), eM-Plant Tutorial, Version 4.6, 2001

UGS Tecnomatix (Hrsg.), eM-Plant Objects Manual, Version 4.0

UGS Tecnomatix (Hrsg.), eM-Plant Reference Manual, Version 4.0

UGS Tecnomatix (Hrsg.), eM-Plant Training Manual 3D, 2002

Internet/Online

(F1) Hilfe Tecnomatix eM-Plant 7.0.10

UGS Hilfecenter [http://www.ugsplm.de/produkte/tecnomatix/plant_design/em_plant.shtml]

Anhang

Bearbeitung in Reihe ungestört

Anzahl WST [St.]	Durchsatz [St.]	Mittlere Durchlaufzeit [s]	Station Arbeit [%]	Station Stoerung [%]	Station Blockiert [%]	Station Warten [%]	Station Leer [%]
0	0	0	0	0	0	100	100
1	261	01:30,0	9,1	0	0	90,9	90,9
2	523	01:30,0	18,19	0	0	81,81	81,81
3	785	01:30,0	27,29	0	0	72,71	72,71
4	1047	01:30,0	36,35	0	0	63,65	63,65
5	1309	01:30,0	45,45	0	0	54,55	54,55
6	1571	01:30,0	54,55	0	0	45,45	45,45
7	1832	01:30,0	63,65	0	0	36,35	36,35
8	2094	01:30,0	72,74	0	0	27,26	27,26
9	2356	01:30,0	81,81	0	0	18,19	18,19
10	2618	01:30,0	90,9	0	0	9,1	9,1
11	2880	01:30,0	100	0	0	0	0
12	2880	01:39,9	100	0	0	0	0
13	2880	01:49,9	100	0	0	0	0
14	2880	01:59,9	100	0	0	0	0
15	2880	02:09,9	100	0	0	0	0
16	2880	02:19,9	100	0	0	0	0
17	2880	02:29,9	100	0	0	0	0
18	2880	02:39,9	100	0	0	0	0
19	2880	02:49,9	100	0	0	0	0
20	2880	02:59,9	100	0	0	0	0
21	2880	03:09,9	100	0	0	0	0
22	2880	03:19,9	100	0	0	0	0
23	2880	03:29,9	100	0	0	0	0
24	2880	03:39,9	100	0	0	0	0
25	2880	03:49,9	100	0	0	0	0
26	2880	03:59,8	100	0	0	0	0
27	2880	04:09,8	100	0	0	0	0
28	2880	04:18,3	100	0	0	0	0
29	2880	04:18,4	100	0	0	0	0
30	2880	04:18,4	100	0	0	0	0
31	2880	04:18,4	100	0	0	0	0
32	2880	04:18,4	100	0	0	0	0
33	2880	04:18,4	100	0	0	0	0
34	2880	04:18,4	100	0	0	0	0
35	2880	04:18,4	100	0	0	0	0
36	2880	04:18,4	100	0	0	0	0
37	2880	04:18,4	100	0	0	0	0
38	2880	04:18,4	100	0	0	0	0
39	2880	04:26,9	100	0	0	0	0
40	2879	04:36,9	100	0	0	0	0
41	2879	04:46,9	100	0	0	0	0
42	2879	04:56,9	100	0	0	0	0
43	2879	05:06,9	100	0	0	0	0
44	2879	05:16,9	100	0	0	0	0
45	2879	05:26,9	100	0	0	0	0

Anzahl WST [St.]	Durchsatz [St.]	Mittlere Durchlaufzeit [s]	Station Arbeit [%]	Station Stoerung [%]	Station Blockiert [%]	Station Warten [%]	Station Leer [%]
46	2879	05:36,9	100	0	0	0	0
47	2879	05:46,9	100	0	0	0	0
48	2879	05:56,9	100	0	0	0	0
49	2879	06:06,9	100	0	0	0	0
50	2879	06:16,9	100	0	0	0	0
51	2879	06:26,9	100	0	0	0	0
52	2879	06:36,9	100	0	0	0	0
53	2879	06:46,9	100	0	0	0	0
54	2879	06:56,9	100	0	0	0	0
55	0		0,03	0	99,97	0	0
56	0		0	0	100	0	0
57	0		0	0	100	0	0
58	0		0	0	100	0	0
59	0		0	0	100	0	0
60	0		0	0	100	0	0

Tabelle 3: *Bearbeitung in Reihe ungestört****Bearbeitung in Reihe gestört (10%)***

Anzahl WST [St.]	Durchsatz [St.]	Mittlere Durchlaufzeit [s]	Station Arbeit [%]	Station Stoerung [%]	Station Blockiert [%]	Station Warten [%]	Station Leer [%]
0	0	0	0	10	0	90	90
1	239	01:39,1	8,3	10	0	81,7	81,7
2	479	01:39,6	16,63	10	0	73,37	73,37
3	719	01:39,7	24,97	10	0	65,03	65,03
4	959	01:39,8	33,3	10	0	56,7	56,7
5	1196	01:40,1	41,53	10	0	48,47	48,47
6	1431	01:40,5	49,72	10	0	40,28	40,28
7	1664	01:40,7	57,78	10	0	32,22	32,22
8	1896	01:41,1	65,83	10	0	24,17	24,17
9	2128	01:41,4	73,89	10	0	16,11	16,11
10	2360	01:41,7	81,94	10	0	8,06	8,06
11	2592	01:41,9	90	10	0	0	0
12	2592	01:53,0	90	10	0	0	0
13	2592	02:04,1	90	10	0	0	0
14	2592	02:15,2	90	10	0	0	0
15	2592	02:26,3	90	10	0	0	0
16	2592	02:37,4	90	10	0	0	0
17	2592	02:48,5	90	10	0	0	0
18	2592	02:59,6	90	10	0	0	0
19	2592	03:10,7	90	10	0	0	0
20	2592	03:21,8	90	10	0	0	0
21	2592	03:32,9	90	10	0	0	0
22	2592	03:44,0	90	10	0	0	0
23	2592	03:54,3	90	10	0	0	0
24	2592	04:04,4	90	10	0	0	0
25	2592	04:14,4	90	10	0	0	0
26	2592	04:24,4	90	10	0	0	0
27	2592	04:34,3	90	10	0	0	0
28	2592	04:42,7	90	10	0	0	0

Anzahl WST [St.]	Durchsatz [St.]	Mittlere Durchlaufzeit [s]	Station Arbeit [%]	Station Stoerung [%]	Station Blockiert [%]	Station Warten [%]	Station Leer [%]
29	2592	04:42,7	90	10	0	0	0
30	2592	04:42,7	90	10	0	0	0
31	2592	04:42,7	90	10	0	0	0
32	2592	04:42,7	90	10	0	0	0
33	2592	04:42,7	90	10	0	0	0
34	2592	04:42,7	90	10	0	0	0
35	2592	04:42,7	90	10	0	0	0
36	2591	04:43,6	90	10	0	0	0
37	2591	04:44,7	90	10	0	0	0
38	2591	04:45,9	90	10	0	0	0
39	2591	04:55,5	90	10	0	0	0
40	2591	05:06,6	90	10	0	0	0
41	2591	05:17,8	90	10	0	0	0
42	2591	05:28,9	90	10	0	0	0
43	2591	05:40,0	90	10	0	0	0
44	2591	05:51,1	90	10	0	0	0
45	2591	06:02,2	90	10	0	0	0
46	2591	06:13,3	90	10	0	0	0
47	2591	06:24,4	90	10	0	0	0
48	2591	06:35,5	90	10	0	0	0
49	2591	06:46,7	90	10	0	0	0
50	2591	06:57,8	90	10	0	0	0
51	2591	07:08,9	90	10	0	0	0
52	2591	07:20,0	90	10	0	0	0
53	2591	07:31,1	90	10	0	0	0
54	2591	07:42,2	90	10	0	0	0
55	0		0,03	10	89,97	0	0
56	0		0	10	90	0	0
57	0		0	10	90	0	0
58	0		0	10	90	0	0
59	0		0	10	90	0	0
60	0		0	10	90	0	0

Tabelle 4: *Bearbeitung in Reihe gestört (10%)*

Parallelbearbeitung Variante 1 ungestört

Anzahl WST [St.]	Durchsatz [St]	Mittlere Durchlaufzeit [s]	Station Arbeit [%]	Station Stoerung [%]	Station Blockiert [%]	Station Warten [%]	Station Leer [%]
0	0	0	0	0	0	100	100
1	250	01:36,0	8,69	0	0	91,31	91,31
2	501	01:36,0	17,39	0	0	82,61	82,61
3	751	01:36,0	26,08	0	0	73,92	73,92
4	1001	01:36,0	34,79	0	0	65,21	65,21
5	1252	01:36,0	43,47	0	0	56,53	56,53
6	1502	01:36,0	52,19	0	0	47,81	47,81
7	1753	01:36,0	60,87	0	0	39,13	39,13
8	2003	01:36,0	69,58	0	0	30,42	30,42
9	2254	01:36,0	78,25	0	0	21,75	21,75
10	2503	01:36,0	86,94	0	0	13,06	13,06
11	2754	01:36,0	95,66	0	0	4,34	4,34
12	2880	01:40,0	100	0	0	0	0
13	2880	01:49,9	100	0	0	0	0
14	2880	01:59,9	100	0	0	0	0
15	2880	02:09,9	100	0	0	0	0
16	2880	02:19,9	100	0	0	0	0
17	2880	02:29,9	100	0	0	0	0
18	2880	02:39,9	100	0	0	0	0
19	2880	02:49,9	100	0	0	0	0
20	2880	02:59,9	100	0	0	0	0
21	2880	03:09,9	100	0	0	0	0
22	2880	03:19,9	100	0	0	0	0
23	2880	03:29,9	100	0	0	0	0
24	2880	03:39,9	100	0	0	0	0
25	2880	03:49,9	100	0	0	0	0
26	2880	03:59,9	100	0	0	0	0
27	2880	04:09,8	100	0	0	0	0
28	2880	04:19,8	100	0	0	0	0
29	2880	04:29,8	100	0	0	0	0
30	2880	04:32,9	100	0	0	0	0
31	2880	04:32,9	100	0	0	0	0
32	2880	04:32,9	100	0	0	0	0
33	2880	04:32,9	100	0	0	0	0
34	2880	04:32,9	100	0	0	0	0
35	2880	04:32,9	100	0	0	0	0
36	2880	04:32,9	100	0	0	0	0
37	2880	04:32,9	100	0	0	0	0
38	2880	04:32,9	100	0	0	0	0
39	2880	04:32,9	100	0	0	0	0
40	2880	04:36,9	100	0	0	0	0
41	2879	04:46,9	100	0	0	0	0
42	2879	04:56,9	100	0	0	0	0
43	2879	05:06,9	100	0	0	0	0
44	2879	05:16,9	100	0	0	0	0
45	2879	05:26,9	100	0	0	0	0
46	2879	05:36,9	100	0	0	0	0
47	2879	05:46,9	100	0	0	0	0
48	2879	05:56,9	100	0	0	0	0
49	2879	06:06,9	100	0	0	0	0
50	2879	06:16,9	100	0	0	0	0
51	2879	06:26,9	100	0	0	0	0
52	2879	06:36,9	100	0	0	0	0
53	2879	06:46,9	100	0	0	0	0
54	2879	06:56,9	100	0	0	0	0

Anzahl WST [St.]	Durchsatz [St.]	Mittlere Durchlaufzeit [s]	Station Arbeit [%]	Station Stoerung [%]	Station Blockiert [%]	Station Warten [%]	Station Leer [%]
55	2879	07:06,9	100	0	0	0	0
56	2879	07:16,9	100	0	0	0	0
57	0		0,06	0	99,94	0	0
58	0		0	0	100	0	0
59	0		0	0	100	0	0
60	0		0	0	100	0	0

Tabelle 5: Ausschleusstation Variante 1 ungestört

Ausschleusstation Variante 1 gestört (10%)

Anzahl WST [St.]	Durchsatz [St.]	Mittlere Durchlaufzeit [s]	Station Arbeit [%]	Station Stoerung [%]	Station Blockiert [%]	Station Warten [%]	Station Leer [%]
0	0	0	0	10	0	90	90
1	231	01:43,3	8,02	10	0	81,98	81,98
2	459	01:44,6	15,97	10	0	74,03	74,03
3	687	01:44,7	23,85	10	0	66,15	66,15
4	912	01:45,4	31,67	10	0	58,33	58,33
5	1136	01:46,1	39,44	10	0	50,56	50,56
6	1360	01:46,5	47,22	10	0	42,78	42,78
7	1584	01:46,8	55	10	0	35	35
8	1808	01:47,0	62,78	10	0	27,22	27,22
9	2032	01:47,2	70,56	10	0	19,44	19,44
10	2256	01:47,3	78,33	10	0	11,67	11,67
11	2480	01:47,4	86,11	10	0	3,89	3,89
12	2592	01:53,0	90	10	0	0	0
13	2592	02:04,1	90	10	0	0	0
14	2592	02:15,2	90	10	0	0	0
15	2592	02:26,3	90	10	0	0	0
16	2592	02:37,4	90	10	0	0	0
17	2592	02:48,5	90	10	0	0	0
18	2592	02:59,6	90	10	0	0	0
19	2592	03:10,7	90	10	0	0	0
20	2592	03:21,8	90	10	0	0	0
21	2592	03:32,9	90	10	0	0	0
22	2592	03:44,0	90	10	0	0	0
23	2592	03:55,2	90	10	0	0	0
24	2592	04:05,4	90	10	0	0	0
25	2592	04:15,5	90	10	0	0	0
26	2592	04:25,6	90	10	0	0	0
27	2592	04:35,5	90	10	0	0	0
28	2592	04:45,4	90	10	0	0	0
29	2592	04:55,3	90	10	0	0	0
30	2592	04:58,3	90	10	0	0	0
31	2592	04:58,3	90	10	0	0	0
32	2592	04:58,3	90	10	0	0	0
33	2592	04:58,3	90	10	0	0	0
34	2592	04:58,3	90	10	0	0	0
35	2592	04:58,3	90	10	0	0	0
36	2592	04:58,3	90	10	0	0	0
37	2591	04:59,2	90	10	0	0	0
38	2591	05:00,3	90	10	0	0	0

Anzahl WST [St.]	Durchsatz [St.]	Mittlere Durchlaufzeit [s]	Station Arbeit [%]	Station Stoerung [%]	Station Blockiert [%]	Station Warten [%]	Station Leer [%]
39	2591	05:01,5	90	10	0	0	0
40	2591	05:06,6	90	10	0	0	0
41	2591	05:17,8	90	10	0	0	0
42	2591	05:28,9	90	10	0	0	0
43	2591	05:40,0	90	10	0	0	0
44	2591	05:51,1	90	10	0	0	0
45	2591	06:02,2	90	10	0	0	0
46	2591	06:13,3	90	10	0	0	0
47	2591	06:24,4	90	10	0	0	0
48	2591	06:35,5	90	10	0	0	0
49	2591	06:46,6	90	10	0	0	0
50	2591	06:57,8	90	10	0	0	0
51	2591	07:08,9	90	10	0	0	0
52	2591	07:20,0	90	10	0	0	0
53	2591	07:31,1	90	10	0	0	0
54	2591	07:42,2	90	10	0	0	0
55	2591	07:53,3	90	10	0	0	0
56	2591	08:04,4	90	10	0	0	0
57	0		0,03	10	89,97	0	0
58	0		0	10	90	0	0
59	0		0	10	90	0	0
60	0		0	10	90	0	0

Tabelle 6: Ausschleusstation Variante 1 gestört (10%)

Ausschleusstation Variante 2 ungestört

Anzahl WST [St.]	Durchsatz [St.]	Mittlere Durchlaufzeit [s]	Station Arbeit [%]	Station Stoerung [%]	Station Blockiert [%]	Station Warten [%]	Station Leer [%]
0	0	0	0	0	0	100	100
1	250	01:36,0	8,71	0	0	91,29	91,29
2	501	01:36,0	17,4	0	0	82,6	82,6
3	751	01:36,0	26,08	0	0	73,92	73,92
4	1001	01:36,0	34,79	0	0	65,21	65,21
5	1252	01:36,0	43,47	0	0	56,53	56,53
6	1502	01:36,0	52,19	0	0	47,81	47,81
7	1753	01:36,0	60,87	0	0	39,13	39,13
8	2003	01:36,0	69,57	0	0	30,43	30,43
9	2253	01:36,0	78,26	0	0	21,74	21,74
10	2504	01:36,0	86,94	0	0	13,06	13,06
11	2754	01:36,0	95,66	0	0	4,34	4,34
12	2880	01:40,0	100	0	0	0	0
13	2880	01:49,9	100	0	0	0	0
14	2880	01:59,9	100	0	0	0	0
15	2880	02:09,8	100	0	0	0	0
16	2880	02:20,0	100	0	0	0	0
17	2880	02:29,7	100	0	0	0	0
18	2880	02:40,1	100	0	0	0	0
19	2880	02:49,3	100	0	0	0	0
20	2880	03:00,4	100	0	0	0	0
21	2880	03:08,5	100	0	0	0	0
22	2881	03:21,3	100	0	0	0	0
23	2879	03:27,5	100	0	0	0	0
24	2880	03:42,4	100	0	0	0	0
25	2880	03:46,7	100	0	0	0	0
26	2881	04:02,9	100	0	0	0	0
27	2879	04:05,7	100	0	0	0	0
28	2881	04:24,0	100	0	0	0	0
29	2879	04:23,3	100	0	0	0	0
30	2881	04:46,5	100	0	0	0	0
31	2879	04:42,3	100	0	0	0	0
32	2881	05:07,4	100	0	0	0	0
33	2879	05:00,3	100	0	0	0	0
34	2881	05:29,5	100	0	0	0	0
35	2880	05:29,8	100	0	0	0	0
36	2879	02:01,3	100	0	0	0	0
37	2880	07:40,1	100	0	0	0	0
38	2879	02:40,7	100	0	0	0	0
39	2880	07:18,9	100	0	0	0	0
40	2880	08:19,9	100	0	0	0	0
41	2879	02:41,0	100	0	0	0	0
42	2880	07:48,8	100	0	0	0	0
43	2880	09:19,9	100	0	0	0	0
44	2880	09:07,9	100	0	0	0	0
45	2880	04:51,1	100	0	0	0	0
46	2880	09:38,4	100	0	0	0	0
47	2880	02:31,3	100	0	0	0	0
48	2880	12:38,3	100	0	0	0	0

Anzahl WST [St.]	Durchsatz [St.]	Mittlere Durchlaufzeit [s]	Station Arbeit [%]	Station Stoerung [%]	Station Blockiert [%]	Station Warten [%]	Station Leer [%]
49	2880	07:49,7	100	0	0	0	0
50	2880	02:41,5	100	0	0	0	0
51	2880	13:28,0	100	0	0	0	0
52	2880	08:19,7	100	0	0	0	0
53	2880	02:51,7	100	0	0	0	0
54	2880	14:17,8	100	0	0	0	0
55	2880	08:48,7	100	0	0	0	0
56	2880	02:57,9	100	0	0	0	0
57	2880	14:59,6	100	0	0	0	0
58	2880	09:08,7	100	0	0	0	0
59	2880	02:58,0	100	0	0	0	0
60	2879	15:40,6	100	0	0	0	0
61	2877	09:10,9	100	0	0	0	0
62	2879	09:21,6	100	0	0	0	0
63	0		0,02	0	99,98	0	0
64	0		0	0	100	0	0
65	0		0	0	100	0	0
66	0		0	0	100	0	0
67	0		0	0	100	0	0
68	0		0	0	100	0	0
69	0		0	0	100	0	0
70	0		0	0	100	0	0

Tabelle 7: Ausschleusstation Variante 2 ungestört

Ausschleusstation Variante 2 gestört (10%)

Anzahl WST [St.]	Durchsatz [St.]	Mittlere Durchlaufzeit [s]	Station Arbeit [%]	Station Stoerung [%]	Station Blockiert [%]	Station Warten [%]	Station Leer [%]
0	0	0	0	10	0	90	90
1	231	01:43,3	8,02	10	0	81,98	81,98
2	459	01:44,6	15,97	10	0	74,03	74,03
3	687	01:44,7	23,85	10	0	66,15	66,15
4	912	01:45,4	31,67	10	0	58,33	58,33
5	1136	01:46,1	39,44	10	0	50,56	50,56
6	1360	01:46,5	47,22	10	0	42,78	42,78
7	1584	01:46,8	55	10	0	35	35
8	1808	01:47,1	62,75	10	0	27,25	27,25
9	2030	01:47,3	70,49	10	0	19,51	19,51
10	2252	01:47,5	78,21	10	0	11,79	11,79
11	2472	01:47,9	85,81	10	0	4,19	4,19
12	2590	01:53,1	89,93	10	0	0,07	0,07
13	2591	02:04,1	89,99	10	0	0,01	0,01
14	2592	02:15,2	90	10	0	0	0
15	2592	02:26,3	90	10	0	0	0
16	2592	02:37,4	90	10	0	0	0
17	2592	02:48,6	90	10	0	0	0
18	2592	02:59,7	90	10	0	0	0
19	2592	03:10,7	90	10	0	0	0
20	2592	03:21,8	90	10	0	0	0
21	2592	03:32,9	90	10	0	0	0

Anzahl WST [St.]	Durchsatz [St.]	Mittlere Durchlaufzeit [s]	Station Arbeit [%]	Station Stoerung [%]	Station Blockiert [%]	Station Warten [%]	Station Leer [%]
22	2592	03:44,0	90	10	0	0	0
23	2592	03:55,1	90	10	0	0	0
24	2592	04:06,2	90	10	0	0	0
25	2592	04:17,3	90	10	0	0	0
26	2592	04:28,4	90	10	0	0	0
27	2592	04:39,6	90	10	0	0	0
28	2592	04:50,6	90	10	0	0	0
29	2592	05:01,8	90	10	0	0	0
30	2592	05:12,8	90	10	0	0	0
31	2592	05:24,0	90	10	0	0	0
32	2592	05:35,1	90	10	0	0	0
33	2592	05:35,3	90	10	0	0	0
34	2592	06:08,3	90	10	0	0	0
35	2592	06:08,4	90	10	0	0	0
36	2592	06:01,1	90	10	0	0	0
37	2592	06:32,2	90	10	0	0	0
38	2592	06:58,7	90	10	0	0	0
39	2592	05:54,3	90	10	0	0	0
40	2592	07:50,0	90	10	0	0	0
41	2592	06:52,0	90	10	0	0	0
42	2592	07:26,2	90	10	0	0	0
43	2592	07:38,5	90	10	0	0	0
44	2592	08:23,1	90	10	0	0	0
45	2592	07:17,4	90	10	0	0	0
46	2592	08:52,7	90	10	0	0	0
47	2592	06:55,6	90	10	0	0	0
48	2592	09:59,0	90	10	0	0	0
49	2592	08:43,9	90	10	0	0	0
50	2592	06:19,7	90	10	0	0	0
51	2592	11:41,4	90	10	0	0	0
52	2592	09:17,2	90	10	0	0	0
53	2592	06:15,3	90	10	0	0	0
54	2592	12:52,5	90	10	0	0	0
55	2592	09:47,5	90	10	0	0	0
56	2592	07:44,7	90	10	0	0	0
57	2592	12:13,9	90	10	0	0	0
58	2592	10:10,9	90	10	0	0	0
59	2592	08:42,4	90	10	0	0	0
60	2586	12:01,3	90	10	0	0	0
61	2588	10:11,1	90	10	0	0	0
62	2591	10:20,7	90	10	0	0	0
63	0		0,04	10	89,96	0	0
64	0		0	10	90	0	0
65	0		0	10	90	0	0
66	0		0	10	90	0	0
67	0		0	10	90	0	0
68	0		0	10	90	0	0
69	0		0	10	90	0	0
70	0		0	10	90	0	0

Tabelle 8: Ausschleusstation Variante 2 gestört (10%)