

Abschlussbericht für das
Bundesministerium für Bildung, Forschung und Technologie (BMBF)

Simulationstoolset für die Schiffbauindustrie

im Rahmenprogramm

Blockbau und Vorausrüstung

vorgelegt von:



Jos. L. Meyer GmbH
(MEYER WERFT)

Postfach 1555
D-26855 Papenburg

Papenburg, im März 2003

Vorhabenbezeichnung:	Simulationstoolset für die Schiffbauindustrie
Zuwendungsempfänger:	Jos. L. Meyer GmbH („JLM“) P.O.Box 1555 26855 Papenburg
Laufzeit des Vorhabens:	01.08.1998 - 30.09.2002
Förderkennzeichen:	18 S 0104 B
Autoren:	Dipl.-Ing. Jörg Hartmann (JLM) Dipl.-Ing. Matthias Krause (JLM) Dipl.-Ing. Edzard Behm (SIPOC GmbH)
Ansprechpartner:	Dipl.-Ing. Jörg Hartmann Fon: +49(4961)81-4577 Fax: +49(4961)81-4497 mailto:hartmann@Meyerwerft.de http://www.meyerwerft.com/
Seiten:	162
Literaturangaben:	17
Bilder:	102
Tabellen:	13

Inhaltsverzeichnis

1	Aufgabenstellung und Bedeutung des Projektes	7
1.1	Ausgangssituation	7
1.2	Ziele	7
1.3	Lösungsansatz	8
1.4	Stand der Technik zu Beginn des Projektes	9
1.5	Organisatorischer Projektablauf	12
1.5.1	Eingesetzte Software	12
1.5.1.1	Auswahlprozess	12
1.5.1.2	Das Simulationssystem eM-Plant	14
1.5.2	Projektphasen (nach Projektplan und Projektverlängerung)	15
1.5.3	Zusammenarbeit und Arbeitsteilung	16
1.5.4	Wissenstransfer und Veröffentlichung	17
2	Fachliche Ergebnisse	18
2.1	Datenstruktur	18
2.1.1	Produktmodell	18
2.1.1.1	Eigenschaftsrepräsentation	19
2.1.1.2	Produkthierarchie	20
2.1.1.3	Topologiesichten	20
2.1.2	Arbeitsplanmodell	21
2.1.3	Ressourcenmodell	21
2.1.4	Realisierung	22
2.1.4.1	Datenbankebene und Simulationsserver	23
2.1.4.2	Simulationsebene	23
2.2	Aufbau des Simulationstoolsets	24
2.2.1	Gestaltungs- und Dokumentationskonzept	24
2.2.1.1	Bausteingestaltung	26
2.2.1.2	Bausteindokumentation	27
2.2.2	Basistoolset	27
2.2.2.1	Dokumentation	29
2.2.2.2	Modellverwaltung	34
2.2.2.3	Fortschritt	35
2.2.2.4	Objektauswahl	36
2.2.2.5	ObjektBackup	38

2.2.2.6	ObjektTransfer	39
2.2.2.7	Search_in_Methods	41
2.2.2.8	Search_for_objects	42
2.2.2.9	Search_in_tables	42
2.2.2.10	TabellenFilter	43
2.2.2.11	Allgemeine Funktionen	45
2.2.3	Datenhaltung und -schnittstellen	46
2.2.3.1	Datenbaustein	46
2.2.3.2	Data-Baustein	53
2.2.3.3	SQL-Baustein	56
2.2.4	Toolset Schiffbauliche Fertigung	62
2.2.4.1	Möglichkeiten und Anwendungsvorteile des Toolsets	62
2.2.4.2	Manager auf Ressourcennetzwerk- und Ressourcenebene	63
2.2.4.3	Allgemeine Steuerungskomponenten	67
2.2.4.4	Ressourcenklassen und ihre Elemente	73
2.2.4.5	Ressourcenklassen	73
2.2.4.6	Öffentliche Methoden der Ressourcenklassen	80
2.2.4.7	Transportmanager	85
2.2.4.8	BEs	87
2.2.4.9	Erstellen von Simulationsmodellen	87
2.2.4.10	Erstellen neuer Toolsetklassen	89
2.2.5	Raumbaustein	90
2.2.5.1	Logische Elemente und Funktionen	90
2.2.5.2	Dialog	91
2.2.5.3	Visualisierung	91
2.2.5.4	Lifecycle und Grundbelegung	92
2.2.6	Übergeordnete Steuerungen	92
2.2.6.1	Pausenbaustein	93
2.2.6.2	Verfügbarkeit	95
2.2.7	Auswertungstools	96
2.2.7.1	Gantt	96
2.2.7.2	BestandGantt	101
2.2.7.3	BestandDiagramm	101
2.2.7.4	Eventschnittstelle	102

2.2.8	Oberflächen	103
2.2.9	Kommunikation.....	106
2.2.9.1	Datenpaket.....	106
2.2.9.2	Kommunikationsbaustein.....	107
2.2.9.3	Datenpaket.....	109
2.2.9.4	Kommunikationsbaustein.....	110
2.3	Datenmanagement.....	112
2.3.1	Architektur	112
2.3.2	SDM Tool.....	112
2.3.2.1	Berechtigungen und Verwaltung.....	112
2.3.2.2	Verwalten von Varianten	113
2.3.2.3	Datenmodell im SDM Tool.....	113
2.3.2.4	Auswertung	117
2.3.3	Modellverwaltung	119
2.3.4	Piloteinsatz	120
2.4	Die Virtuelle Werft.....	121
2.4.1	Fertigungsplanung (Anwendungsebene).....	122
2.4.1.1	Sim_Schedule.....	122
2.4.1.2	Sim_Space (Raumbelegung)	126
2.4.2	Fabrikplanung (Expertenebene)	129
2.4.2.1	Grobsimulation.....	129
2.4.2.2	Panelstraße.....	130
2.5	Unterstützende Verfahren	135
2.5.1	CAD Datentransfer.....	135
2.5.1.1	Anforderungen	135
2.5.1.2	Realisierung.....	135
2.5.2	Datengenerierung	136
2.5.2.1	Workflow	137
2.5.2.2	Realisierung.....	138
2.5.3	Optimierung	139
2.5.3.1	Implementierte Verfahren	139
2.5.3.2	Definition der Lösungsgüte (Fitness).....	140
2.5.3.3	Optimierungsverfahren.....	141
	Ergebnisse und Vergleich.....	146

2.5.4	Bausteinverwaltung und Dokumentation	148
2.5.4.1	Anforderungen „Dokumentation“	149
2.5.4.2	Bausteinverwaltung	149
2.6	Sonstige Arbeitspunkte	153
3	Zusammenfassung	154

1 Aufgabenstellung und Bedeutung des Projektes

1.1 Ausgangssituation

Der Wettbewerb im internationalen Schiffbau verschärft sich auch in Folge des erheblichen Preisverfalls bei asiatischen Werften ständig. Davon sind inzwischen auch klassische Domänen des deutschen und europäischen Schiffbaus, wie z.B. Container- und Kreuzfahrtschiffe, betroffen. Die deutsche Schiffbauindustrie wird in Zukunft nur am Markt bestehen können, wenn sie u.a. durch gezielte Forschung und Entwicklung hochwertige Produkte nach dem neuesten Stand der Technik zu wettbewerbsfähigen Preisen mit kurzen Lieferzeiten anbieten kann. Dabei ist es wichtig, auch bis kurz vor Fertigstellung des Schiffes noch flexibel auf Kundenwünsche und technische Veränderungen reagieren zu können.

Wesentlichen Einfluss auf Produktivität, Lieferzeit und Flexibilität haben die Gestaltung der Abläufe in Entwurf und Fertigung sowie die effektive Steuerung der Material- und Informationsflüsse auf strategischer, taktischer und operativer Ebene. Mit zunehmender Komplexität des Produktes steigt die Bedeutung von Planung und Logistik für die effektive Nutzung der Ressourcen einer Werft. Die Meyer Werft in Papenburg hat diese Zusammenhänge erkannt und in den letzten Jahren erhebliche Entwicklungsleistungen erbracht. Gerade auf den Gebieten Planung und Logistik gehört sie heute zu den führenden Werften der Welt.

Dabei hat sich jedoch gezeigt, dass konventionelle Planungstechniken mit zunehmender Komplexität und Vernetzung der Abläufe an ihre Grenzen stoßen. So stellt die realitätsnahe Berücksichtigung der ständigen Veränderungen in Produkt und Fertigung mit der traditionellen Netzplantechnik ein grundsätzliches Problem dar. Gefragt sind Werkzeuge, die es ermöglichen, Fertigungsvarianten mit geringem Aufwand zeitdynamisch vor auszudenken. Um kostenintensive "Experimente" und Fehlentscheidungen in der Realität zu vermeiden, weicht man sowohl bei der Investitionsplanung (Fabrikplanung), als auch bei der alltäglichen Fertigungsablaufplanung immer mehr auf rechnergestützte Planungstechniken aus. Moderne Planungsmethoden wie die Simulation und neue Methoden der Informationsverarbeitung wie Expertensysteme, Fuzzy-Logic, neuronale Netze und Optimierungsstrategien, müssen in die traditionellen Planungstechniken eingebunden werden, um die Planungsqualität in Zukunft weiter zu verbessern.

In verschiedenen Projekten, auch auf der Meyer Werft, wurden Voruntersuchungen angestellt und Simulationsmodelle für Insellösungen erstellt. Trotzdem steckt die Simulation im Schiffbau noch in den Kinderschuhen. So ist es bisher nicht gelungen, die Werftabläufe in ihrer Gesamtheit in die Simulation einzubinden und neue Planungstechniken auch Simulationslänen auf allen Entscheidungsebenen der Werft zugänglich zu machen. Unzureichende Programmierungstechniken führen dazu, dass die Modellierung der "Werft im Rechner" der sich dynamisch verändernden Realität "hinterherhinkt" und so keine optimalen Ergebnisse erzielt werden.

1.2 Ziele

Das hier beschriebene Projekt hat deshalb nicht das primäre Ziel, den vorhandenen Simulationsmodellen weitere werftspezifischen Insellösungen hinzuzufügen. Vielmehr sollen die Grundvoraussetzungen für den effektiven Einsatz der Simulation im Schiffbau allgemein deutlich verbessert werden, indem:

- der Aufwand zur Modellerstellung durch einen modular aufgebauten “Simulationswerkzeugkasten” reduziert und damit die Simulation näher an die Realität gebracht wird,
- die Nutzerfreundlichkeit verbessert und damit die Simulation aus dem “Elfenbeinturm” der Simulationsexperten herausgebracht und Praktikern auf Anwendungsebene als Planungswerkzeug erschlossen wird,
- neue Techniken der Informationsverarbeitung in den Werkzeugkasten eingebunden werden und damit auch solche schiffbautypischen Problemkreise realitätsnah modelliert werden können, wie das Wissen der Meister zur Steuerung der Abläufe auf operativer Ebene oder die zeitdynamische Nutzung von Flächenressourcen,
- durch die Pilotanwendung des Bausteinkastens auf der Meyer Werft der Nachweis erbracht wird, daß es mit dem entwickelten Werkzeugkasten möglich ist, Werftabläufe auch in der gesamten Komplexität und nicht nur als Insellösungen mit vertretbarem Aufwand zu simulieren.

Eine besondere Bedeutung des Projektes liegt darin, daß durch Nutzung einer kommerziellen Simulationssoftware als Basis für den Toolset und aufgrund seines modularen Aufbaus die Ergebnisse des Projektes und die Erfahrungen der Meyer Werft auch anderen interessierten Werften direkt zugänglich sind.

1.3 Lösungsansatz

Zur Erreichung der genannten Ziele gliedert sich das Projekt in drei Aufgabenkomplexe, die wie in Abbildung 1 dargestellt zusammenwirken und aufeinander aufbauen.

Der Aufgabenkomplex 1 - *Erstellung des Simulationstoolsets* umfasst die Entwicklung der Softwarebausteine für den Toolset auf Basis der ausgewählten Simulationssoftware einschließlich der dazu notwendigen Analyse, Strukturierung und Modellierung schiffbautypischer Prozesse und Abläufe. Daneben beinhaltet dieser Aufgabenkomplex die Integration der Simulationsbausteine und weiterer unterstützender Tools (Aufgabenkomplex 3). Im Gegensatz zur Pilotanwendung des Toolsets für die spezifischen Gegebenheiten der Meyer Werft im Aufgabenkomplex 2 steht in AK 1 die schiffbauspezifische, aber werftübergreifende Ausrichtung des Toolsets im Mittelpunkt.

Die Pilotanwendung und den Test des Toolsets auf der Meyer Werft umfasst der Arbeitskomplex 2 - *Die virtuelle Werft*. Dazu gehört sowohl die Nutzung der Einzelbausteine aus dem Werkzeugkasten zur Modellierung einzelner Unternehmensbereiche, als auch die Zusammenführung der Module zum Gesamtmodell Werft.

Schließlich dient Arbeitskomplex 3 - *Modulare Tools zur Unterstützung der Simulation* dazu, Lösungen für wiederkehrende Standardprobleme der Planung durch den Einsatz moderner Methoden der Informationsverarbeitung zu entwickeln. Der Arbeitskomplex umfasst grundlegende Aufgabenstellungen, die nicht unbedingt mit der kommerziellen Simulationssoftware gelöst werden. Die entwickelten Module sollen sowohl vollintegriert im Simulationstoolset, als auch unabhängig von diesem als Stand-Alone-Lösungen in der Planung einsetzbar sein.

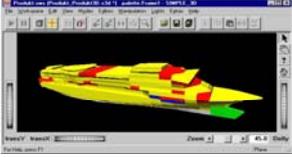
<p>1. Simulationstoolset</p> <p>Modellierungstool </p> <p>Visualisierungstool </p> <p>Auswertungstool </p> <p>2. Die virtuelle Werft</p>  <p>3. Analytische Verfahren</p> <p> Optimierung</p> <p> Statistische Versuchsplanung</p>	<p><u>Entwicklungszeitraum :</u> 09.1098-09.2002</p> <p><u>Ziele :</u></p> <ul style="list-style-type: none"> • Aufbau eines Simulations-“Toolsets“ • Modellierung von Fertigungsbereichen (virtuelle Werft) <p><u>Vorteile :</u></p> <ul style="list-style-type: none"> • Erhöhte Planungssicherheit bei der Fabrik- und Fertigungsablaufplanung • verbesserte Ressourcennutzung • Fehlerverminderung 	
<p>02.12.02 12:07</p>	<p>Projektziele / Aufgabenkomplexe</p>	

Abbildung 1 - Projektziele/Aufgabenkomplexe

1.4 Stand der Technik zu Beginn des Projektes

Nach VDI-Richtlinie 3633 [2] ist die Simulation ein Verfahren zur Nachbildung von Systemen mit ihren dynamischen Prozessen in einem experimentierbaren Modell, mit dem Ziel, zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind. Sie wird insbesondere dort eingesetzt, wo die Komplexität von Prozessen und Abläufen analytische Optimierungsmethoden an ihre Grenzen stoßen lässt. Die Simulation basiert heute fast ausschließlich auf Modellen im Rechner und wird zusammen mit anderen computergestützten Techniken (CAD, CAPE, CAM) eingesetzt.

Voraussetzung für den erfolgreichen Einsatz der Simulation sind hardwareseitig leistungsfähige Computer (Pentium-PC oder Workstation), die heute zu relativ moderaten Preisen zur Verfügung stehen.

Die softwareseitigen Voraussetzungen wurden in den letzten Jahren ständig weiterentwickelt. Simulationswerkzeuge können nach [4] in drei Generationen eingeteilt werden:

Die erste Generation von Simulationssoftware basiert auf der textuellen Modellierung durch Programmiersprachen (z.B. C++, FORTRAN, SLAM etc.). Diese Softwarewerkzeuge sind zwar hinsichtlich der Modellerstellung sehr flexibel, d.h. es können beliebige Probleme und Abläufe modelliert werden, der Aufwand zur Modellerstellung ist aber extrem hoch.

Die zweite Generation basiert auf der parametrischen Modellierung durch spezielle Simulationssoftware (z.B. WITNESS, DOSIMIS etc.). Mit Systemen der zweiten Generation können spezielle Aufgabenstellungen modelliert und simuliert werden. Der Aufwand für die Modellerstellung wurde in dieser Generation erheblich vermindert, jedoch mit dem Nachteil, dass die Flexibilität eingeschränkt wurde und nur mit dem vorhandenen Pool vorgefertigter Elemente gearbeitet werden kann. Die Simulation von Problemen, die branchenspezifisch für den

G:\Abschluss\Bericht\Gesamtbericht v03.11.doc

Schiffbau sind, ist mit Simulationsprogrammen der zweiten Generation oft nur unzureichend möglich (vgl. dazu Erfahrungen der Meyer Werft beim Einsatz von WITNESS).

Die dritte Generation von Simulationssoftware basiert auf der objektorientierten, hierarchischen Modellierung (z.B. Simple++, ARENA etc.). Mit dieser Generation können auch spezielle Fragestellungen mit vertretbarem Aufwand modelliert werden. Durch die offene Struktur können speziell entwickelte Programme in die Simulation eingebunden und integrierte Systeme geschaffen werden. Ohne die Vorfertigung von Standardbausteinen ist der Programmieraufwand "vor Ort" jedoch auch bei Systemen der dritten Generation zu hoch, um den Ansprüchen an eine zeitnahe Modellierung zu genügen. Außerdem ist zur Erstellung der Modelle Spezialwissen erforderlich, das auf der Anwenderebene kaum vorhanden ist. Der Trend bei der Anwendung von Simulationssystemen geht hin zu objektorientierten offenen Systemen der dritten Generation.

1.4.1 Einsatz der Simulation in der Serienfertigung

Simulationswerkzeuge werden seit mehreren Jahren im Bereich der Automobilindustrie, der chemischen Industrie, aber auch im Dienstleistungsbereich, z.B. in Banken eingesetzt. Wenig verbreitet ist die Simulation dagegen in Bereichen mit stark auftragsorientierter Einzelfertigung, wie dem Großanlagenbau, im Bauwesen und im Schiffbau.

Hinsichtlich des Einsatzes der Simulation in der Serienfertigung ergeben sich folgende entscheidende Unterschiede zum Schiffbau:

- Die Simulation wird vorwiegend bei der Investitionsplanung eingesetzt. Im Gegensatz zum Schiffbau werden Fertigungslinien einmal sorgfältig geplant, danach läuft die Fertigung, gesteuert von "konventionellen" Planungstechniken nach dem optimierten Muster ab. Die Auslegung neuer Fertigungslinien liegt zeitlich meist weit vor der Fertigung, so dass ausreichend Zeit für die Erstellung der Simulationsmodelle zur Verfügung steht. Da es sich bei der Investitionsplanung im wesentlichen um eine "einmalige" Aufgabe handelt, ist ein hoher Programmieraufwand eher vertretbar als unter ständig veränderlichen Bedingungen wie im Schiffbau.
- Im Bereich der Fertigungsplanung treten bei einer Serienfertigung auf weitgehend automatisierten bzw. getakteten Linien wesentlich weniger "Störgrößen" auf, als im Schiffbau. Sich ständig verändernde Produkte, der Einfluss des "Faktors Mensch" in manuellen Montagebereichen und die flexible Belegung von Produktionsflächen sind Probleme, die z.B. im Automobilbau kaum auftreten.
- Für die Hersteller von Simulationssoftware stellt die Serienfertigung in der Regel einen weitaus interessanteren Markt dar, als "exotische" Bereiche, wie der Schiffbau. Erst in letzter Zeit ist aufgrund der Sättigung der Märkte für Simulationssysteme im Bereich der Großserienfertigung ein gewisses Umdenken zu verzeichnen. Der Schiffbau kann also nicht darauf vertrauen, dass ohne sein Zutun ein optimales Simulationssystem für seine Belange entwickelt wird, sondern muss hier Eigeninitiative entwickeln.

Um den Programmieraufwand zu senken, wurden auch für Serienfertiger, wie den Automobilbau und die chemische Industrie, bereits Bausteinpakete entwickelt. Speziell angepasste Simulationssysteme für kleinere Industriezweige wie die Werftindustrie sind zur Zeit auf dem Markt nicht verfügbar.

1.4.2 Stand der Planung und Simulation auf der Meyer Werft

Die Meyer Werft besitzt ein ausgereiftes Planungs- und Managementinformationssystem, das in den letzten Jahren durch die Werft selbst auf Basis des "Application Systems" (AS) der Firma IBM entwickelt wurde. AS stellt eine Art höhere Programmiersprache für Planungs- und Projektmanagementaufgaben dar. Durch einen Anwendungsprogrammierer für AS wurden auf der Meyer Werft Planungsbausteine geschaffen, die von den Planern der Abteilungen für speziell auf deren Anforderungen zugeschnittenen Anwendungen genutzt werden. In diesem Sinne wurde im Bereich der Planung bereits eine Art Bausteinsystem entwickelt, das sich jedoch im wesentlichen auf den Einsatz netzplangestützter Techniken beschränkt.

Die Daten, die über Jahre im vorhandenen Planungssystem gespeichert werden, stellen eine sehr gute Basis für die Simulation dar. Neben reinen Planungsdaten können aus der EDV der Meyer Werft weitere für die Simulation benötigte Informationen, wie Stücklisteninformationen, CAD - Daten und Arbeitszeitinformationen gewonnen werden. Die vorhandene Datenbasis, die modulare Struktur des Planungssystems einschließlich realisierter Schnittstellen z.B. zur Materialverwaltung und die Erfahrungen mit einem Bausteinsystem sind Ergebnisse, auf die bei der Erstellung des Simulationstoolsets aufgebaut werden kann.

1.4.3 Werftspezifisches CIM-Konzept

Im Zusammenhang mit der Entwicklung des Planungs- und Managementinformationssystems wurde auf der Meyer Werft vor einigen Jahren eine umfassende Analyse der Werftabläufe sowie der Material- und Informationsflüsse durchgeführt. Ziel dieses Projektes war die Erstellung eines betriebsspezifischen CIM-Konzeptes, das im EDV-System der Meyer Werft (Materialverwaltung, Planung, Anbindung des Brennschneidzentrums) zum großen Teil auch umgesetzt wurde.

Die Prozessanalysen aus dem betrieblichen CIM-Konzept können heute als Basis für die Modellierung bestimmter Grob Abläufe mit Hilfe der Simulation genutzt werden. Eine Erfahrung des CIM-Projektes war es jedoch auch, dass sich die Produkte und die Fertigungsabläufe schneller veränderten, als die Entwicklung der CIM-Werkzeuge, insbesondere der Planungsinstrumente, vorangetrieben werden konnte. Diese Erkenntnis fand in der Konzeption des hier beschriebenen Werkzeugkastens ihre Berücksichtigung.

1.4.4 Einsatz der Simulation auf der Meyer Werft

Zur Unterstützung der Planung nutzt die Meyer Werft seit etwa vier Jahren die Simulation. Bisher werden mit der Simulationssoftware WITNESS [3] bzw. in der Programmiersprache C++ erstellte Modelle sowohl bei der Investmentplanung, als auch zur Fertigungsablaufplanung genutzt.

Einige Beispiele zur Anwendung der Simulation auf der Meyer Werft seien im folgenden genannt:

- Investitionsplanung
 - Untersuchung verschiedener Baufolgen im Baudock
 - Investitionsvorbereitung Brennschneidzentrum
 - Prozessvisualisierung zur Fehlersuche Brennschneidzentrum
 - Untersuchung Auslegungsvarianten Paneelstraße
 - Bestimmung von Taktzeiten für eine Laserschweißanlage für Sandwichpaneele

- Simulation der Weiterverarbeitung von Sandwichpaneelen zu Decks
- Fertigungsablaufplanung
 - Simulation der Flächenbelegung im Blockbau (ständig eingesetzt)
 - Kapazitätsplanung im Technischen Büro Schiffbau (ständig eingesetzt)
 - Simulationsmodell zur Verifizierung der Planungsgrößen in der Vorfertigung
 - Simulationsmodell zur Tagesprogrammplanung im Brennschneidzentrum
 - Modell zur operativen Planung im Bereich Paneelstraße (in Vorbereitung)
- Sonstiges
 - Simulationsmodell zur Auslegung von Fluchtwegen auf Kreuzfahrtschiffen

Simulationsmodelle für den Bereich der Fertigungsablaufplanung wurden durch spezielle Programme voll in die Planungsumgebung integriert (Start der Simulation aus der Planungsumgebung heraus, Rückgabe der Ergebnisse und Auswertung im Planungstool). Aufgrund der genannten Ergebnisse und nach zahlreichen Gesprächen mit anderen Werften im In- und Ausland kann davon ausgegangen werden, dass die Meyer Werft führend bei der Simulation von Werftabläufen ist.

Bisherige Erfahrungen der Meyer Werft mit der Simulation haben aber auch gezeigt, dass viele werftspezifische Problemstellungen mit kommerziellen Systemen nicht bzw. nur mit einem erheblichen Aufwand realisiert werden können. Die Modellerstellung dauert zu lange, und es ist nicht gelungen, die Simulation in alle Planungsebenen zu integrieren.

1.5 Organisatorischer Projektablauf

Mit Beginn des Projektes Ende 1998 waren bereits einige Weichen für das Projekt gestellt. Dies betraf vor allem die für die Programmierung und Modellierung benötigten Werkzeuge. Diese wurden im Vorfeld des Projekts ausgewählt.

1.5.1 Eingesetzte Software

Als Datenbank wurde für die erste Projektphase Microsoft Access ausgewählt. Als Auswahlkriterium stand eine möglichst einfache Bedienung und Modellierung auf der Datenbankseite im Vordergrund. Ein Migration auf ein leistungsfähigeres Datenbanksystem sollte im Projekt dennoch möglich bleiben. Für Oberflächen (GUI) für den Anwender wurde Visual Basic ausgewählt, da auf der Meyer Werft bereits umfangreiches „Know How“ mit dieser Programmiersprache zur Verfügung stand. Als Simulationssoftware wurde eM-Plant von Tecnomatix in einem umfangreichen Benchmark ausgewählt.

1.5.1.1 Auswahlprozess

Unter den auf dem Markt zur Zeit angebotenen Systemen wurden in einer Grobauswahl vier Simulationssysteme ausgewählt, die den von der Meyer Werft gestellten Grundkriterien (z.B. Anwendungsgebiet, Schnittstellen, Systemumgebung, Software-Handhabung etc.) entsprachen. Um diese Systeme weiter zu differenzieren und besser bewerten zu können, sollen anhand eines prägnanten Beispielmodells (Abbildung 2) die Stärken und Schwächen, bezogen auf die Anwendung im schiffbaulichen Bereich, der einzelnen Systeme verdeutlicht werden. Gleichzeitig wird das vorhandene Entwicklungspotential abgeschätzt.

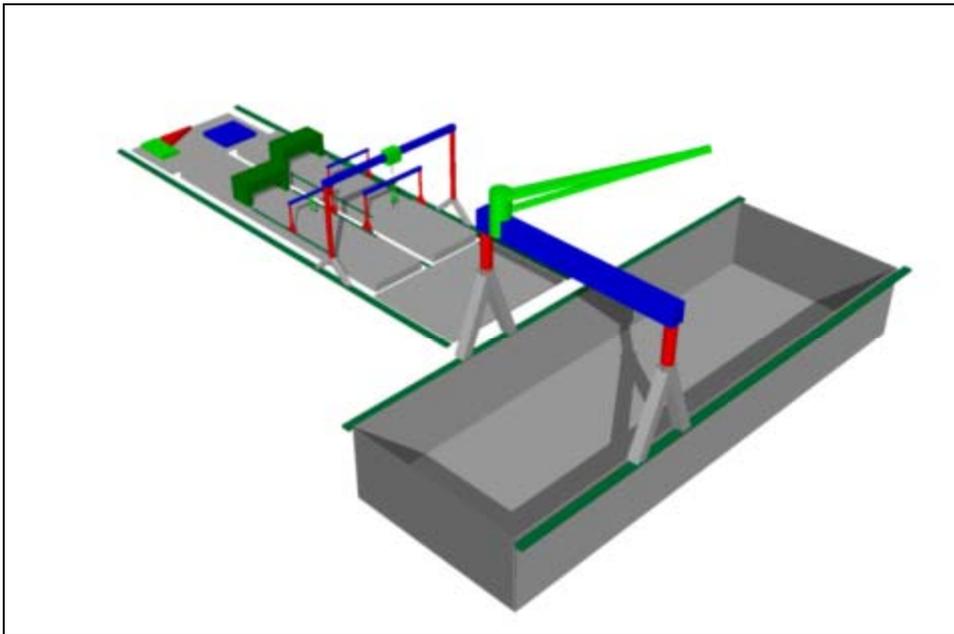


Abbildung 2 - Software-Auswahlprozess: Benchmark-Modell

Die Bewertungstabelle zeigt eine Zusammenstellung der Ergebnisse des Benchmark-Modells der Meyer Werft. Die Bewertung wurde nach folgenden Gesichtspunkten durchgeführt:

- ++ sehr gute Lösung der Aufgabenstellung
- + eine Lösung für die Aufgabenstellung wurde aufgezeigt (neutrale Bewertung)
- schlechte Lösung der Aufgabenstellung

Anforderungen:	eM-Plant	
Chaotisches Lager	Freies Lager über Raster	++
Schiffbauspezifische Maschinen	Bewegliche Maschinen	+
Krane	Portalkrane und Schwenkkran	+
Teile	Wurden modelliert	+
Betriebskalender	Wurden hinterlegt	+
User-Interface	Dialogbox über Schnittstelle	+
Interaktive Eingriffe	Sind möglich	++
Montageplan	Steuerlogik wurde modelliert	+
Grafische interaktive Auswertung	Interaktive Gantt Charts	++
Darstellungshierarchien	Hohe Detailgenauigkeit	+
Grafische Darstellung	2D-Pixelgrafik	--
Modellalternativen	Experimente	-+
Optimierung	Beispiel für GA-Optimierer	+

Tabelle 1 - Kriterien und Bewertung des eM-Plant Benchmark-Modells

G:\Abschluss\Bericht\Gesamtbericht v03.11.doc

Nach Abwägung aller Vor- und Nachteile wurde für das Projekt „Simulationstoolset für die Werftindustrie“ die Simulationssoftware eM-Plant von Tecnomatix ausgewählt.

1.5.1.2 Das Simulationssystem eM-Plant

Die wesentlichen Eigenschaften der Simulationssoftware sind nach Darstellung von Tecnomatix [<http://em-plant.de/>]

„...die vollständige Objektorientierung von eM-Plant:

Grafische Benutzeroberfläche, Systemarchitektur und Implementierung entsprechen den Kennzeichen der Objektorientierung. Somit stehen dem Benutzer Eigenschaften wie etwa Hierarchie, Vererbung, Polymorphismus, etc. zur Verfügung. Ebenso wichtig wie die Eigenschaften der Objektorientierung sind die Kommunikations- und Integrationsfähigkeit von eM-Plant. Es stehen Schnittstellen zur Echtzeitkommunikation mit anderer Software, wie etwa ODBC- oder SQL-Datenbanken und PPS-Systemen, zur Verfügung. Damit kann eM-Plant als integrierte Simulationskomponente eines umfassenden Informationssystems verwendet werden. eM-Plant ist auf den Plattformen Windows NT und Windows 2000 lauffähig.“

Im Hinblick auf die Projektziele sind folgende Features von eM-Plant hervorzuheben:

- Objektorientierung, Vererbung
- Hierarchischer Aufbau von Objekten und Modellen durch Schachteln von Netzwerken
- Bausteinbibliothek-Konzept mit Standard- und benutzerdefinierten Bausteinen
- Bausteine können separat in Dateien gespeichert werden.
- Spezialisierte Bausteinkästen v.a. für Automobilindustrie erhältlich
- Simulation von Fertigungs- und Geschäftsprozessen, Materialflusssimulation
- Optimierung mit Genetischen Algorithmen
- Grafische und tabellarische Analyse- und Statistiktools (Engpassanalyse, Gantt Wizard, Sankey Diagramme u.a.)
- Programmierung spezieller Logiken in benutzerdefinierten Methoden in der Sprache SimTalk; Debugger und Profiler: umfassende, flexible Informationsverarbeitung, Hilfen zur Fehlerbeseitigung und Modelloptimierung
- Erstellung benutzerdefinierter Dialogmasken für den Anwender
- Separate und integrierte 3D-Animation
- Offene Architektur mit etlichen Schnittstellen (ASCII, ActiveX, C, COM, DDE, HTML, ODBC, SQL, Socket etc.)
- Experimentverwalter zur automatisierten Simulation und Auswertung von Experimentreihen

1.5.2 Projektphasen (nach Projektplan und Projektverlängerung)

Das Projekt “Simulationstoolset für den Schiffbau” gliedert sich in drei Aufgabenkomplexe. Der Zusammenhang zwischen diesen Arbeitskomplexen wurde in Abbildung 1 dargestellt und im Abschnitt 1.3 erläutert. Die Arbeitskomplexe laufen aufgrund der vielfältigen Wechselbeziehungen weitgehend parallel ab. Erstellte Simulationsbausteine können so schnell in werft-spezifische Modelle umgesetzt und getestet werden.

AK 1: *Erstellung des Simulationstoolsets* beinhaltet

- alle konzeptionellen und strukturierenden Aufgaben,
- die Erstellung von Anwendungsbausteinen auf der Basis von eM Plant
- die Integration der Anwendungsbausteine und der unter AK3 erstellten unterstützenden Tools,
d.h. die *Erstellung des schiffbauspezifischen Toolsets*

AK 2: *Werft im Rechner* beinhaltet

- die Anwendung des Toolsets (AK1) auf die konkreten Gegebenheiten der Meyer Werft mit dem Ziel, die praktische Realisierbarkeit und Funktion der Bausteine zu testen,
- die werftspezifischen Datenanbindungen mit Hilfe der Servicebausteine (AK1),
- die Schaffung von Modellen für die Meyer Werft,
d.h. die *konkrete Pilotanwendung*

AK 3: *Unterstützende Softwaretools* beinhaltet

- die Entwicklung derjenigen Bausteine für den Simulationstoolset, die nicht mit eM-Plant, sondern unter Nutzung anderer Methoden modelliert werden,
d.h. die *Zulieferung von Sonderbausteinen* für AK1.

Einige Bausteine konnten gleich zu Beginn der Projektes umgesetzt werden. Für andere Bausteine war eine längere Konzeptphase erforderlich, bevor mit der Umsetzung begonnen werden konnte. Viele Bausteine und Objekte wurden während der gesamten Projektlaufzeit gepflegt, gewartet und weiterentwickelt. Ein erheblicher Teil der Projektarbeit bestand in der Pflege der erstellten Bausteine. Dadurch wurde sichergestellt, dass im Projekt eine vollständig integrierte Bausteinbibliothek erstellt wurde.

In der ersten Projektphase zeigte sich sehr schnell, dass die Modellierung von komplexen Steuerungen im Modell noch mit erheblichen Schwierigkeiten verbunden ist, da diese Algorithmen nur im Modell hinterlegt und vom Simulationsexperten gewartet und gepflegt werden können. Um die Anwendbarkeit durch simulationsunkundiges Personal (Planer etc.) zu verbessern, muss die Modellierung der Steuerung durch eine definierte Datenstruktur innerhalb einer bedienerfreundlichen “Front End” Lösung ermöglicht werden. Dadurch wird die Modellierung von Steuerungen zu einer reinen Datenmodellierung, die dann auch von Planern etc. ausgeführt werden kann. Diese Vorgehensweise ist aus Sicht des Projektteams der Schlüssel für die erfolgreiche Umsetzung der Simulationsstrategie. Auf der Basis dieser Ergebnisse und der neuen Strategie für das Gesamtprojekt wurde eine Projektverlängerung und ein modifizierter Arbeitsplan eingereicht und vom Projektträger genehmigt.

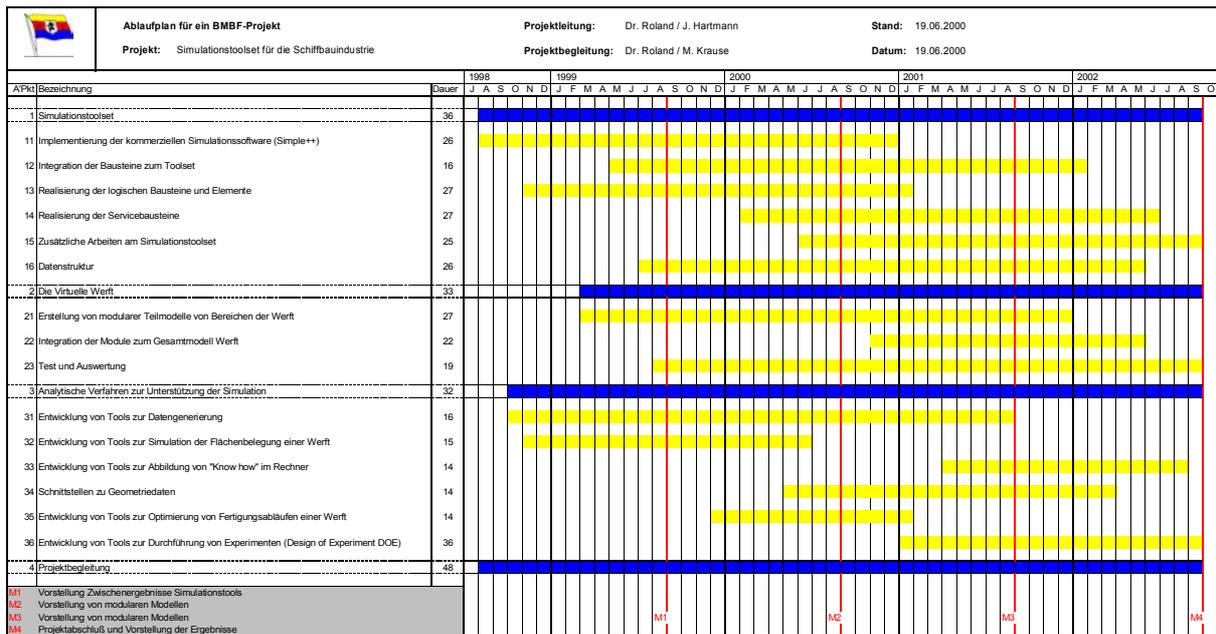


Abbildung 3 - Modifizierter Arbeitsplan

Durch die neue Strategie wurden die Themenschwerpunkte für das Projektteam neu definiert und aufgeteilt. Ein Themenschwerpunkt war nun der Aufbau eines leistungsfähigen Datenmodells, der entsprechenden Datenbankumgebung (Clint-Server) und der entsprechenden „Front End“ Applikationen. Der zweite Themenschwerpunkt war die Umsetzung der Simulationsstrategie in eM-Plant. Dazu gehörte die Konzeption der Bausteine, die Entwicklung von Schnittstellen zum Datenmodell und der Aufbau von Modellen auf Basis der entwickelten Bausteine.

Bei der Einführung der Simulation (Prototypen) auf der Meyer Werft zeigte sich sehr schnell, dass mit dem Toolset ein Simulationssystem geschaffen wurde, mit dem Planer in ihrem Planungsprozess stark unterstützt werden. Durch die konsequente Umsetzung der Strategie der Datenmodellierung kann der Planer Abläufe aus seiner Sicht modellieren, ohne dass vertiefte Simulationskenntnisse erforderlich sind.

1.5.3 Zusammenarbeit und Arbeitsteilung

Das Projekt „Simulationstoolset für die Werftindustrie“ wurde von der Jos. L. Meyer GmbH (Meyer Werft) beantragt und durchgeführt. Dafür wurde eine Förderung durch das Bundesministerium für Bildung und Forschung (BMBF) in Anspruch genommen. Den zuständigen Stellen beim BMBF und bei den Projektträgern Schiffstechnik des BMBF – zunächst Germanischer Lloyd, seit Mai 2001 Forschungszentrum Jülich PTJ – sei an dieser Stelle für die Unterstützung und die sachdienlichen Hinweise gedankt. Eine erfolgreiche Durchführung des Projektes wäre ohne die fachliche Unterstützung unterschiedlicher Partner aus Forschung und Industrie nicht möglich gewesen. Die Zusammenarbeit mit diesen Partnern wurde in Abstimmung mit dem Projektträger durch Industrieaufträge realisiert, die im Ergebnis von Ausschreibungen nach dem Preis-Leistungs-Verhältnis vergeben wurden. Gleichzeitig wurde mit den Partnern eine langfristige Zusammenarbeit angestrebt, um ein gemeinsames Verständnis für das Projekt und die fachliche Problematik zu entwickeln.

Insbesondere ist an dieser Stelle der Firma Tecnomatix GmbH, Stuttgart, und der Sipoc GmbH, Bremen, für vielfältige Unterstützung und fachliche Diskussionen gedankt.

Bei der Abwicklung des Projektes war nicht zuletzt eine Vielzahl von Mitarbeitern der Meyer Werft, insbesondere der Projektteams und der Planungsabteilung, direkt oder indirekt beteiligt. Auch Ihnen sei an dieser Stelle gedankt.

1.5.4 Wissenstransfer und Veröffentlichung

Die Meyer Werft hat die Ergebnisse des Forschungsprojektes „Simulationstoolset für die Werftindustrie“ während der gesamten Projektlaufzeit dem Projektträger, interessierten Werften und auf Fachtagungen präsentiert. Beispielhaft sind im folgenden einige der Präsentationen genannt:

- 05. 1999 Gemeinsame Durchführung des Seminars: „Simulation in der Werftindustrie“ mit Tecnomatix in Hamburg.
- 09. 1999 Erster Meilenstein : Vorstellung der Projektergebnisse, Papenburg
- 10. 1999 Vortrag zum Thema „Erstellung eines Simulationstoolsets für die Werftindustrie auf der Basis von eM-Plant“, Tecnomatix Anwendertreffen, Stuttgart
- 05. 2000 Erfahrungsaustausch zum Thema Simulation bei der Flensburger Schiffbau-Gesellschaft.
- 05. 2001 Zweiter Meilenstein: Vorstellung der Projektergebnisse.
- 06. 2001 Workshop „Simulation“ mit der Kvaerner Warnow Werft auf der Meyer Werft.
- 12. 2001 Vortrag zum Thema „Simulationswerkzeuge für die Schiffbauindustrie“ auf dem STG Sprechtag „Simulation und Modellierung“.
- 03. 2002 Erfahrungsaustausch zum Thema Simulation mit der Flensburger Schiffbau-Gesellschaft auf der Meyer Werft.
- 07. 2002 Vortrag zum Thema „Einsatz der Simulation zur Unterstützung der operativen Produktionsplanung im Schiffbau“ auf dem Tecnomatix Anwendertreffen, Stuttgart

Folgende Veröffentlichungen von Projektergebnissen wurden bereits durchgeführt:

- /1/ Konstruktionspraxis 2000/5, „Computertechnologien im modernen Schiffbau“, Seiten 26-27
- /2/ Produktion und Automatisierung 2000/4 „Simulationstool unterstützt die Planungslogistik beim Bau von Traumschiffen“, Seiten 28-29
- /3/ Tagungsband eM-Plant Benutzertreffen 2002

Eine weitere Veröffentlichung zum Thema „Einsatz der Simulation zur Unterstützung der operativen Produktionsplanung im Schiffbau“ ist für das zweite Halbjahr 2003 geplant.

2 Fachliche Ergebnisse

In den ersten Unterkapiteln werden das entwickelte Simulationstoolset und die flankierenden dem Datenmanagement dienenden Tools detailliert beschrieben (2.1 - 2.3).

Es folgt eine Auswahl der im Rahmen des Teilprojekts „virtuelle Werft“ erstellten Modelle, anhand derer die praktischen Einsatzmöglichkeiten des Toolsets veranschaulicht werden (2.4).

Das Kapitel schließt mit einem Abriss über die entwickelten zusätzlichen Verfahren und Tools, die im Rahmen des Projekts neben dem Simulationstoolset entwickelt wurden.

2.1 Datenstruktur

In diesem Abschnitt werden die drei Säulen Produktmodell - Arbeitsplanmodell - Ressourcenmodell vorgestellt, auf denen die Datenmodellierung für die Simulation fußt. Diverse Tools werden genannt, die zur Unterstützung der simulationsgerechten Datenmodellierung entwickelt wurden und in den anschließenden Kapiteln dokumentiert sind.

Für die Durchführung einer Simulationsstudie sind die Daten die wichtigste Grundlage. Für die meisten Simulationsstudien werden Daten aus verschiedenen Bereichen benötigt, wie z.B. Produktdaten aus der Konstruktion, Arbeitsplan und Prozessdaten aus der Planung und Ressourcendaten aus der Stammdatenverwaltung. Im Projekt wurde daher für die Simulation ein Datenmodell entwickelt, indem Produktdaten, Arbeitsplandaten und Ressourcendaten in einzelnen Komplexen abgebildet werden. Abbildung 3 zeigt das entwickelte Datenmodell und die logische Aufteilung. Bei allen Überlegungen stand im Vordergrund, dass im Schiffbau die Fertigung von Unikaten bzw. Kleinstserien vorherrschen. Dieser Anforderung musste im Datenmodell dementsprechend Rechnung getragen werden. Eine interaktive Modellierung der Daten in allen drei Komplexen ist somit für den Planer unbedingt erforderlich. Das Datenmodell musste erweiterbar sein, damit neue Anforderungen sofort in das bestehende Datenmodell integriert werden können.

Schnittstellen zu Fremdsystemen sollten mit standardisierter Technologie wie. z.B. ODBC ermöglicht werden, damit diese Systeme flexibel an die Simulation angebunden werden können.

2.1.1 Produktmodell

Die Entwicklung des allgemeinen Produktmodells für die Simulation im Schiffbau wurde von folgenden Hauptgedanken bestimmt:

- Datenreduzierung: Welche Bauteileigenschaften sind im Hinblick auf die Anforderungen einer Materialflusssimulation relevant?
- Geometrie: Welche Detaillierung ist bei der Abbildung der Bauteilgeometrie erforderlich?
- Hierarchie: Wie wird die durch den Montagebaum vorgegebene Hierarchie flexibel abgebildet?
- Topologiesichten: Wie können die in verschiedenen Werftbereichen vorherrschenden unterschiedlichen Topologiesichten auf das gleiche Produkt in einem gemeinsamen Modell berücksichtigt werden?

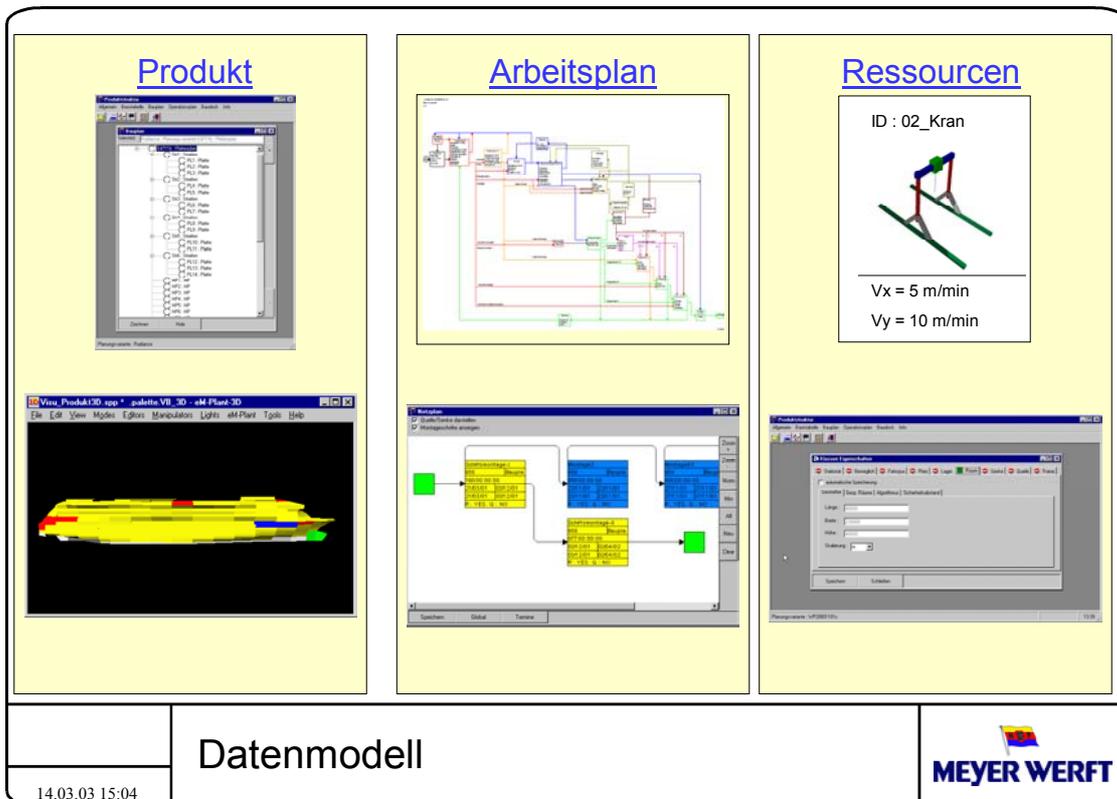


Abbildung 3 - Datenmodell

Für die folgenden Bauteileigenschaften ist zunächst die simulationstechnische Relevanz zu erläutern:

- Bauteilgestalt: Die zu handhabenden Bauteile besitzen eine unregelmäßige Formgestaltung, die nicht über einfache Formen abgebildet werden kann.
- Bauteilausdehnung: Die Bauteile besitzen in allen drei Raumrichtungen eine nicht zu vernachlässigende Ausdehnung.

2.1.1.1 Eigenschaftsrepräsentation

Um die Bauteileigenschaften im Simulationssystem abbilden zu können, wurde eine vereinfachende Repräsentationsform gewählt:

- Zur Bauteilgeometrie:

Die Bauteilgeometrie wird mit einer 8-Eckpunkte-Logik beschrieben. Zu jedem Element der Produktstruktur werden die Koordinaten der Eckpunkte eines umhüllenden Hexaeders gespeichert. Die Gestalt kann von der in Abbildung 4 gezeigten Quaderform abweichen – es sind auch windschiefe Formen und Prismen möglich. Damit ist die Bauteilgestalt in einer für die Materialflusssimulation hinreichenden Form beschrieben. Die Koordinaten sind stets auf den definierten Nullpunkt (Nullspant) im Heck des Schiffs bezogen (Abbildung 5) und beschreiben somit gleichzeitig die Einbauposition des Bauteils.

Die aus den Koordinaten errechneten Hauptausdehnungen des Bauteils werden als Bauteilattribute gespeichert. Den Ausdehnungen in x-, y- und z-Richtung des Schiffs sind die Attribute Länge, Breite und Höhe begrifflich fest zugeordnet.

- Weitere Bauteilattribute wie Massen, Materialinformationen etc. sollen im Datenmodell hinterlegt werden können, damit diese Attribute für verschiedene Simulationen genutzt werden können.

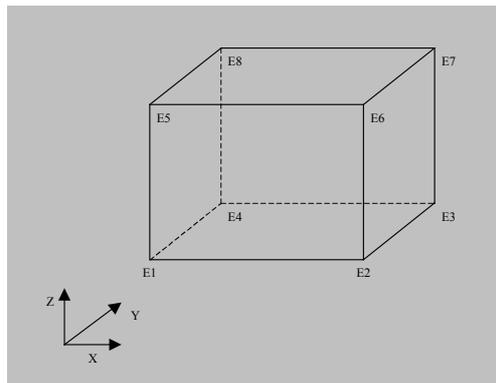


Abbildung 4 - Bauteilkoordinatenverwaltung

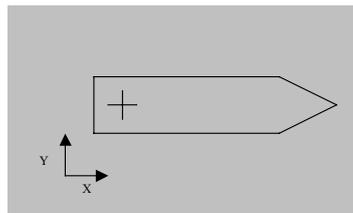


Abbildung 5 - Schiffskordinaten

2.1.1.2 Produkthierarchie

Für die Materialflusssimulation des Fertigungsprozesses (Stahlbau) ist die Hierarchy of Assembly (Montagebaum) zu beachten. Eine standardisierte Abstufung wie z.B. Schiff / Block / Sektion / Einzelteil ist nicht praktikabel, zumal das Produktmodell in den verschiedenen Phasen der Konstruktion, Arbeitsablaufplanung und Fertigung verschiedene Änderungen durchleben kann, in denen Blöcke zu Modulen und Bauabschnitten zusammengefasst oder Sektionen in Teilsektionen zerlegt werden müssen.

Ein Produktmodell „Stahl“ muss daher geeignet sein, die Struktur des Produkts Schiff flexibel abzubilden. Zudem sollen erforderliche Änderungen im Datenmodell mit vertretbarem Aufwand nachvollzogen werden können.

2.1.1.3 Topologiesichten

Sobald über den stahlbaulichen Fertigungsprozess hinausgehend weitere Werftbereiche simuliert werden sollen, geraten andere Sichtweisen auf das Produkt Schiff ins Blickfeld. Sowohl Konstruktionsabteilungen als auch der Schiffsausrüstung ist die Unterteilung des Schiffs in Decks und Feuerzonen von Bedeutung, für andere Gewerke spielt die Einteilung der wasserdichten Abteilungen eine herausragende Rolle.

Ein umfassendes Produktmodell für den Schiffbau vereinigt in sich die Abbildung aller relevanten Sichten und kann diese zueinander in Beziehung setzen (z.B. welche Sektionen liegen in einer Feuerzone).

2.1.2 Arbeitsplanmodell

Im Arbeitsplan werden die in der Simulation auszuführenden Arbeitsschritte gespeichert. Die spezielle Situation einer Schiffswerft als Einzel- oder Kleinserienfertiger machte es erforderlich, eine Lösung zu finden, die über die Modellierung von an Bauteiltypen orientierten Typarbeitsplänen hinausgeht. Im Datenmodell wurden individuelle Arbeitspläne implementiert, in denen zu jedem Einzelbauteil innerhalb einer Planungsvariante sämtliche Arbeitsschritte explizit aufgeführt werden.

Arbeitspläne enthalten folgende grundlegende Informationen:

- **Arbeitsschritte**

Der lokale Arbeitsplan für ein Bauteil besteht aus einem oder mehreren Arbeitsschritten. Ein Arbeitsplan beginnt vor dem ersten echten Arbeitsschritt immer mit dem Schlüsselwort „Quelle“ und endet nach dem letzten echten Arbeitsschritt mit dem Schlüsselwort „Senke“ (beide Schlüsselworte nicht zu verwechseln mit den gleichnamigen Ressourcen). Die Definition von Montageschritten erlaubt die Verknüpfung der Arbeitspläne im Sinne der Hierarchy of Assembly.

Zwar besteht der Gesamt-Arbeitsplan einer Planungsvariante meist aus einer Anzahl lokaler Arbeitspläne, die jeweils auf ein Bauteil bezogen sind. Aufgrund besonderer Arbeitsstrukturen in einigen Bereichen der Werft (Konstruktionsbüros) wurde jedoch auch die Möglichkeit geschaffen, Arbeitsschritte, die auf unterschiedliche Bauteile bzw. Topologien bezogen sind, miteinander zu verknüpfen. Ein weiteres Merkmal des Arbeitsplanmodells ist die Möglichkeit, Arbeitspläne zu hierarchisieren. Jedem Arbeitsschritt kann ein Arbeitsplan untergeordnet sein.

- **Prozesse**

Jedem Arbeitsschritt ist ein Prozess zugewiesen. Prozesse sind als grundlegende zeitverbrauchende Aktionseinheit definiert. Zur Ausführung von Prozessen sind ein oder mehrere Qualifikationen notwendig. Diese werden von den Ressourcen zur Verfügung gestellt. Transportvorgänge werden aufgrund ihrer anders gearteten Struktur nicht wie Prozesse abgebildet.

- **Verknüpfungen**

Die Verknüpfungen der Arbeitsschritte entscheiden über die Reihenfolge ihrer Abarbeitung. Das Datenmodell ermöglicht die Modellierung verschiedener Logiken:

- sequentielle Verknüpfungen: einem Arbeitsschritt ist stets ein Nachfolger zugeordnet (Standardverhalten, im Toolset implementiert)
- alternative Verknüpfungen: ein Arbeitsschritt hat mehrere mögliche Nachfolger, zur Simulationszeit wird entschieden, welcher der alternativen Arbeitsschritte auszuführen ist (Spezialverhalten, im Toolset grundsätzlich implementiert, Entscheidungsmethoden müssen für den jeweiligen Anwendungsfall programmiert werden)
- parallele Arbeitsschritte: ein Arbeitsschritt hat mehrere zeitlich auszuführende Nachfolger (Spezialverhalten, als Prototyp modelliert)

2.1.3 Ressourcenmodell

Ressourcen können Bauteile aufnehmen und transportieren (z.B. Ressourcenklassen Platz, Container) und / oder die für die Ausführung von Prozessen benötigten Qualifikationen (z.B.

Ressourcenklassen Maschinen, Werker) anbieten und so aktiv an der Ausführung von Arbeitsschritten teilnehmen.

Zum einen muss im Ressourcenmodell das Vorhandensein der Ressourcen an sich abgebildet sein. Zudem ist es erforderlich, die verschiedenen Eigenschaften und Fähigkeiten der Ressourcen zu modellieren. Unterschieden werden spezielle Eigenschaften, die jeweils nur für eine bestimmte Ressourcenklasse sinnvoll sind, und allgemeine Ressourceneigenschaften, die verschiedenen Ressourcenklassen attribuiert werden. Spezielle Eigenschaften sind für Krane z.B. der Arbeitsraum, oder für Hallen ihre geometrischen Abmessungen. Allgemeine Eigenschaften sind z.B. für alle Ressourcen der jeweilige zugewiesene Schichtkalender.

Ressourcen können in Modellen organisiert werden. Innerhalb von Modellen können Ressourcen gruppiert und zugeordnet werden. Werker können dadurch zu Teams gruppiert, Krane bestimmten Hallen und Bereichen zugeordnet werden.

2.1.4 Realisierung

Die Umsetzung dieses Konzeptes im Toolset erforderte Aktivitäten in drei Bereichen. Das Datenmodell musste entwickelt und auf der Datenbankebene umgesetzt werden (vgl. 1.1.4.1). Für die Eingabe von Daten musste eine Applikation erstellt werden, mit der alle Daten bearbeitet werden können (vgl. 2.3). Des weiteren mussten Schnittstellen zu Fremdsystemen geschaffen werden, um eine Übernahme von Produktionsdaten gewährleisten zu können. Auf der Simulationsebene musste das Datenmodell in Bausteinen und Objekten abgelegt werden (vgl. 2.2). Abbildung 6 zeigt, wie ausgehend vom Datenmodell (SQL Server) unterschiedliche Simulationsapplikationen mit Daten aus dem integrierten Datenmodell versorgt werden.

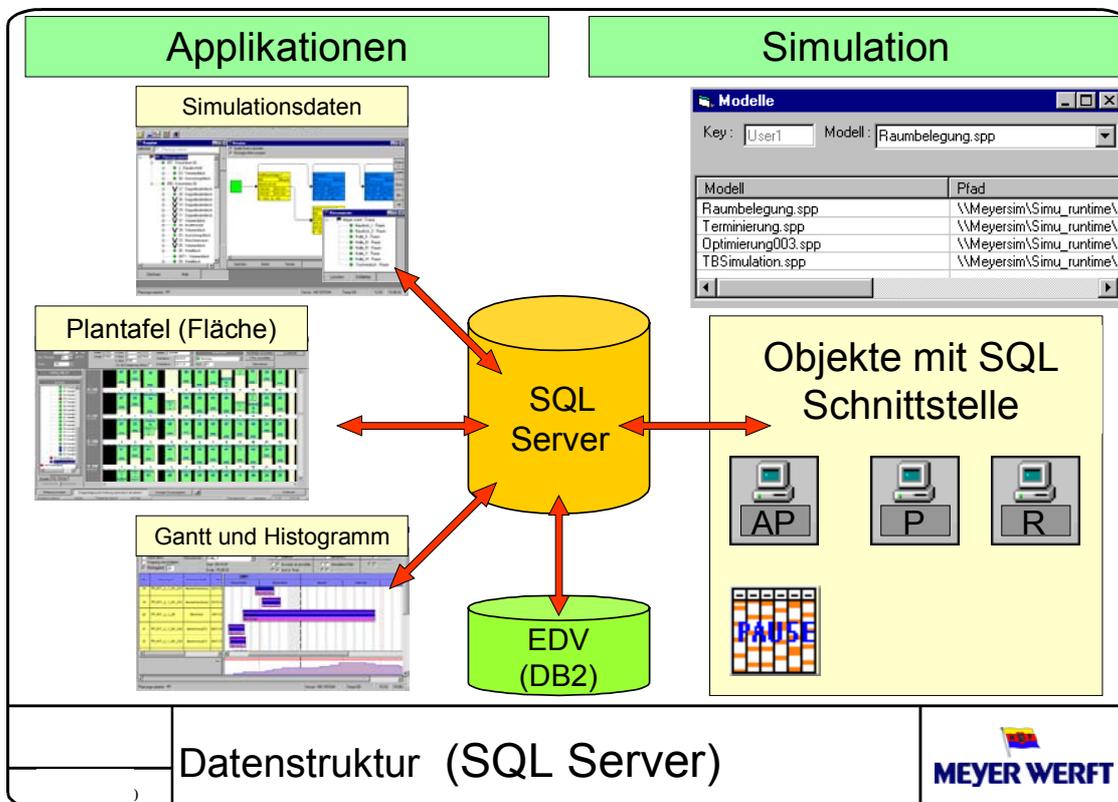


Abbildung 6 - Umsetzung Datenstruktur

2.1.4.1 Datenbankebene und Simulationsserver

Die Umsetzung des Datenmodells wurde auf der Datenbankebene und auf der Simulationsebene realisiert. Die Umsetzung auf der Datenbankebene wurde im ersten Schritt mit Microsoft Access (File Server) realisiert. In zwei Phasen wurde ein relationales Datenmodell entwickelt, mit dem die drei Säulen Produkt, Arbeitsplan und Ressourcen in der Datenbank abgebildet werden konnten. In der ersten Phase wurde das Produkt abgebildet, in der zweiten die Arbeitspläne und Ressourcen. Während beider Phasen wurde das Datenmodell getestet und für erste Simulationsanwendungen verwendet. Die frühzeitige Umsetzung des Konzeptes erlaubte die gezielte Weiterentwicklung und Pflege des Datenmodells während des gesamten Projektes. Alle entwickelten Simulationskomponenten konnten gezielt auf das Datenmodell abgestimmt werden.

In einem weiteren Entwicklungsschritt wurde eine Migration der Access Datenbank auf den SQL Server durchgeführt. Die Migration war erforderlich, damit die erforderliche Performance gewährleistet werden konnte und dem Planer eine Multiuser-Umgebung zur Verfügung gestellt werden konnte. Mit der Migration wurde für das gesamte Datenmanagement eine Client-Server-Architektur (Zweischicht-Architektur) realisiert. Die Migration konnte mit geringem Aufwand realisiert werden, da schon zu Beginn der Projektes eine spätere Umstellung auf ein leistungsfähiges Datenbanksystem geplant war.

Für die „Toolset“-Entwicklung wurde ein Server (Simulationsserver) auf der Meyer Werft installiert. Auf dem Server wurden alle im Projekt entwickelten Bausteine und Objekte, alle Modelle und die Simulationsdatenbank abgelegt. Eine zentrale Verwaltung der Objekte und der Modelle wurde so im Projekt gewährleistet.

Insgesamt konnte durch die Realisierung auf der Datenbankebene ein vollständig in die Simulation integriertes Datenmodell entwickelt werden, das die Grundlage für alle Simulationen im Projekt ist.

2.1.4.2 Simulationsebene

Die Umsetzung des Datenkonzeptes in der Simulation erforderte einige konzeptionelle Arbeit, bevor mit der Umsetzung begonnen werden konnte. Die entsprechenden Bausteine und Objekte wurden spezifiziert und umgesetzt. Die Repräsentanz der drei Säulen auf Simulationsseite sind für das Produktmodell der sogenannte „Data“-Baustein und für das Arbeitsplan- und das Ressourcenmodell der sogenannte „Datenbaustein“. Sowohl die Namensgebung als auch die Tatsache, dass zwei unterschiedliche Bausteine für die Abbildung eines zusammengehörenden Datenmodells entwickelt wurden, können aus der zeitlichen Entwicklung des Toolsets heraus erklärt werden. Eine Zusammenführung der beiden Bausteine zu einem „großen“ Datenbaustein, der die drei Hauptsäulen der Modellierung von Simulationsdaten repräsentiert, wird angestrebt. Beide Bausteine sind auf der Simulationsebene elementare Werkzeuge, die in jedem Modell verwendet werden.

Die ausführliche Dokumentation des Data- und des Datenbausteins wurde in den Abschnitt „Aufbau des Simulationstoolsets“ integriert (2.2.3.1, 2.2.3.2).

2.2 Aufbau des Simulationstoolsets

Dieses Kapitel ist in folgende Abschnitte gegliedert:

- Vorangestellt ist eine Beschreibung von Konzepten und Methoden, die bei der Gestaltung und Dokumentation des Toolsets eingesetzt wurden (2.2.1). Es folgt die ausführliche Dokumentation der einzelnen Bausteine:
- Basistoolset: Sammlung nicht schiffbauspezifischer Bausteine, die die von eM-Plant zur Verfügung gestellten Standardfunktionen ergänzen.
- Datenhaltung und -schnittstellen: Tools, mit denen simulationsrelevante Daten in eM-Plant zur Verfügung gestellt werden. Neben der Grundstruktur und -funktionalität werden auch die wesentlichen Merkmale der Datenstruktur beschrieben, auf der das Verhalten der in den darauffolgenden Abschnitten behandelten Bausteine beruht.
- Toolset Schiffbauliche Fertigung: Bausteine, in denen unterschiedliche Simulationslogiken modelliert sind. Dazu gehören allgemeine Ablaufsteuerungen und schiffbauspezifische Fertigungsressourcen
- Raumbaustein: Spezialbaustein, der die besonderen Erfordernisse der Engpassressource Fläche/Raum auf einer Kompaktwerft abbildet.
- Übergeordnete Steuerungen: weitere Komponenten für die Simulationsablaufsteuerung.
- Auswertungstools: statistische Auswertung und Visualisierung, Reaktion auf Simulationsergebnisse
- Oberflächen: Allgemeine und spezielle Dialogfunktionen, die auf ActiveX basieren
- Kommunikation: Objekte, über die der Informationsaustausch der Komponenten des Toolsets untereinander geführt wird.

2.2.1 Gestaltungs- und Dokumentationskonzept

Zu Beginn des Projekts wurden Richtlinien für die Gestaltung und Dokumentation der Komponenten vereinbart. Im Laufe des Projekts wurden spezielle Werkzeuge erworben bzw. entwickelt, die die Komponenten zu dokumentieren und zu verwalten helfen. Damit wurden mehrere Ziele verfolgt:

- optimale Handhabung und Überschaubarkeit für den Entwickler
- Verkürzung der Einarbeitungszeit für neue Entwickler
- Wiedererkennbarkeitswert für den Anwender
- Schnittstellen zwischen den Bausteinen bzw. nach extern

Dem Konzept des Toolsets folgend erstrecken sich die Richtlinien auf drei Ebenen:

- Methoden und andere Basiselemente
 - Namenskonventionen: Gewissen Methodennamen sind bereits durch eM-Plant Funktionen fest zugeordnet, z.B. werden alle Methoden mit dem Namen „Init“ unmittelbar vor Simulationsstart ausgeführt. Für das Toolset wurden zusätzliche Namens-

konventionen vereinbart, die die Aufgabe von Methoden und Variablen verdeutlichen oder auch die Methodenausführung bei bestimmten Ereignissen bewirken.

- Farbkonventionen: Die Farbe, mit der globale Variablen im Netzwerk dargestellt werden, wird durch ihre Aufgabe bestimmt (z.B. durch Anwender parametrierbar, fest eingestellte Pfade oder Konstanten, Zählvariablen)
- Symbolkonventionen: Die von eM-Plant zur Verfügung gestellten Icons, mit denen Methoden im Netzwerk dargestellt werden, heben ihre jeweilige Funktion hervor, z.B. ein grünes „M“ für öffentliche Methoden. Dem Toolset wurden noch einige Symbole hinzugefügt, z.B. das nebenstehende Symbol „UserDefinable“ für Methodenrumpfe, die lediglich vorformuliert sind und für den jeweiligen Anwendungsfall ausformuliert werden müssen. 
- Methodendokumentation: Zur Methodendokumentation wird der „MethodManager“ der Firma Tecnomatix verwendet (Abbildung 7). Er ermöglicht dem Programmierer, neu angelegte bzw. geänderte Methoden sofort zu dokumentieren. Die Dokumentation wird in den Kopfzeilen des Methodentextes gespeichert. Außerdem bietet der Dokumentationsbaustein (2.2.2.1) die Möglichkeit, mit Hilfe des MethodManagers Dokumentationen bestimmter Methoden eines Bausteins im HTML-Format zu generieren.

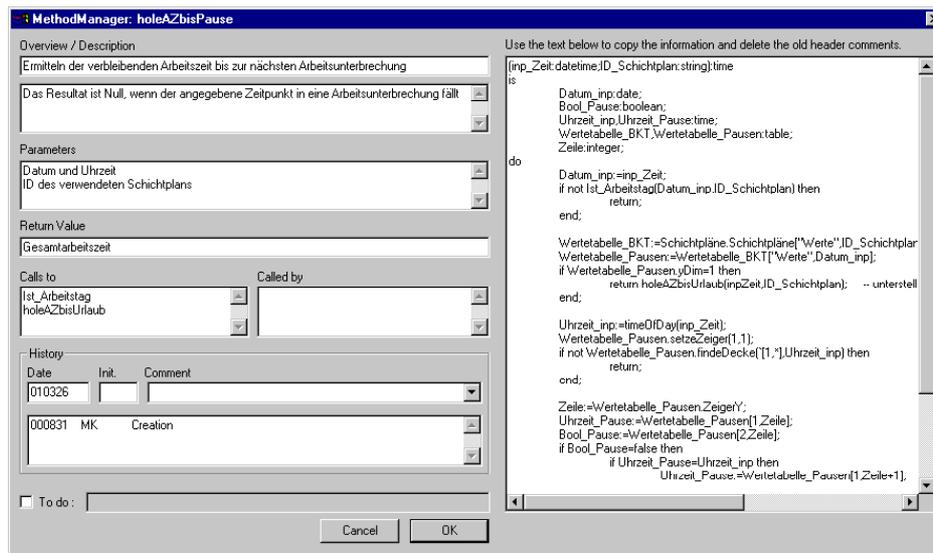


Abbildung 7 - MethodManager: Eingabemaske

- Bausteine: Sämtliche Gestaltungsregeln, Dokumentations- und Verwaltungstools für Simulationsbausteine wurden vollständig von der Meyer Werft konzipiert und realisiert. Es sind dies:
 - Bausteindokumentation innerhalb der Simulationsumgebung (2.2.2.1)
 - Datenbank, in der Bausteindokumentationen erfasst werden
 - Anwendung „Bausteinverwaltung“ (vgl. 2.5.4.2)

Da das Regelwerk für die Objektgestaltung als ein Kernstück des Toolsetentwurfs anzusehen ist, folgt weiter unten eine eingehendere Beschreibung. Die entwickelten Tools werden an den angegebenen Orten erklärt.

- Modelle: Da jedes erstellte Simulationsmodell individuell auf spezielle Fragestellungen ausgerichtet ist und daher die Methoden ihrer Dokumentation wenig formalisierbar sind, wurden für die Modelldokumentation keine speziellen Tools oder Datenbanken entwickelt. Ausführliche Dokumentationen werden in Form von Handbüchern und/oder Powerpoint-Präsentationen erstellt. Daneben besteht im Tool „Bausteinverwaltung“ die Möglichkeit anzugeben, in welchen Modellen und bestimmte Bausteine des Toolsets eingesetzt sind.

2.2.1.1 Bausteingestaltung

Um die Erstellung neuer Anwenderbausteine zu erleichtern, wurde ein eM-Plant-Objekt als Formatvorlage erstellt. Alle Komponenten der Klasse befinden sich in einem gemeinsamen Hauptordner (Abbildung 8-A). Grundsätzlich enthält jeder Baustein zwei Unterordner:

- Ordner „Basisklassen“ (elementare eM-Plant-Objekte, die zum Aufbau des Objekts erforderlich sind) und
- Ordner „Hilfsskassen“: komplexere Objekte - abgeleitet aus Basisklassen - in denen Teilfunktionen des Bausteins modelliert sind.

Zusätzlich befinden sich im Hauptordner mehrere Netzwerke, in denen - abgeleitet aus Basis- und Hilfsklassen - die Hauptfunktionen des Bausteins modelliert sind (Abbildung 8):

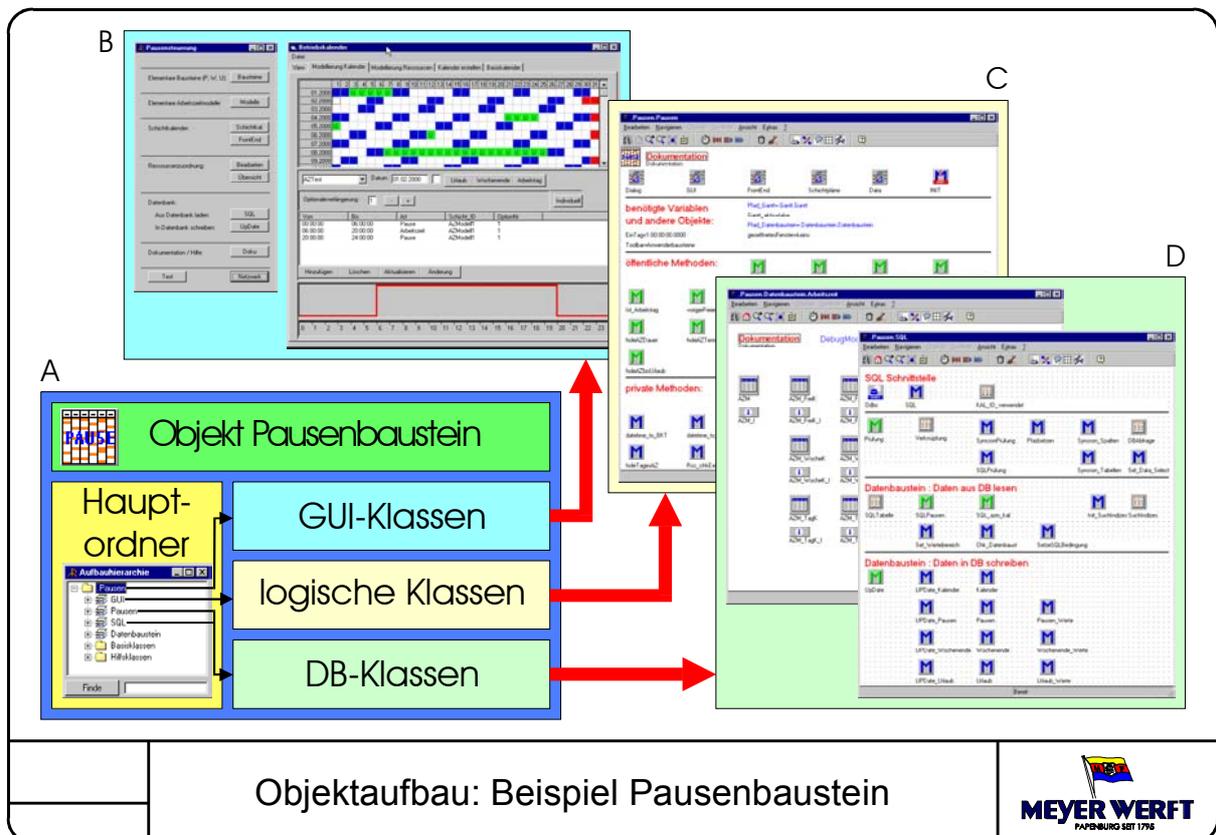


Abbildung 8 - Objektaufbau: Beispiel Pausenbaustein

- „GUI-Klasse“: Das Netzwerk enthält Dialoge und ActiveX-Steuerungen, die für die Steuerung und/oder Parametrierung des Bausteins benötigt werden.

- „logische Klasse“: Die logische Klasse eines Bausteins - auch Fachklasse genannt - ist das Netzwerk, in dem die eigentliche Funktionalität modelliert ist. Meist ist es gleichzeitig dasjenige Netzwerk, das in ein Simulationsmodell instanziiert werden kann.
- „Datenbank-Klassen“: Viele Bausteine verfügen über eine Datenbankschnittstelle. Die Funktionalitäten sind aufgeteilt auf einen Datenbaustein (2.2.3.1), in den die eingelesenen oder zu schreibenden Daten gespeichert werden, und ein SQL-Netzwerk (2.2.3.3), das die Methoden für den Lese- und Schreibzugriff auf die Datenbank enthält

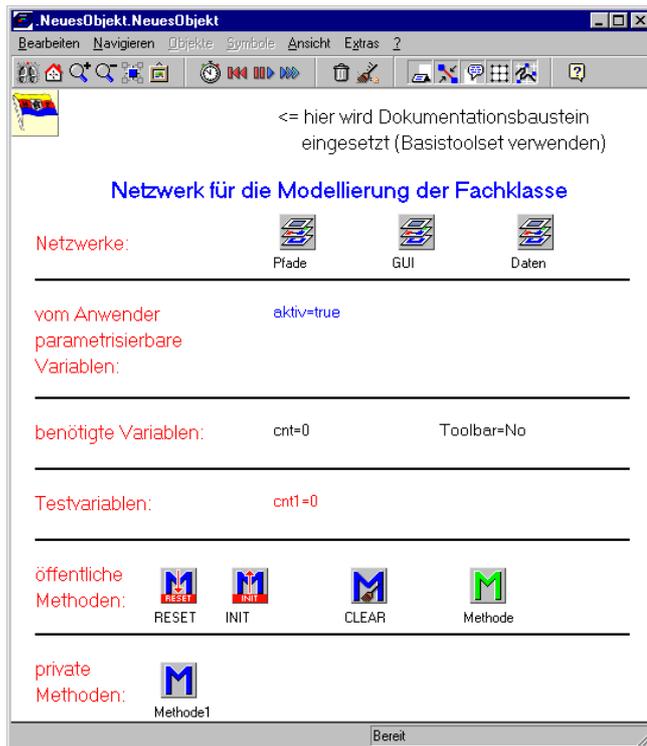


Abbildung 9 - Objektgestaltung: Formatvorlage für Anwenderbausteine

2.2.1.2 Bausteindokumentation

Zur Dokumentation von Anwenderbausteinen stehen zwei Tools zur Verfügung:

Der Dokumentationsbaustein (2.2.2.1) dient als eM-Plant-Objekt der Speicherung von Informationen, die ein Objekt allgemein beschreiben. Daneben werden Anwenderbausteine in einer Datenbank zentral dokumentiert.

- schnelles Finden von Bausteinen aufgrund von Schlagwörtern
- Dokumentation von Modellen (Welcher Baustein ist in welchem Modell eingesetzt)

2.2.2 Basistoolset

Konzept und Aufbau

Neben den Toolsetobjekten, die im Laufe des Projekts für spezielle Simulationsaufgaben entwickelt wurden, entstanden etliche Bausteine allgemeinerer Art. Da auf diese besonders häufig und von den verschiedensten Objekten des Toolsets zugegriffen wird, wurde ein Ba-

sistoolset eingeführt, das die am häufigsten verwendeten allgemeinen Bausteine in sich vereinigt. Dies hat folgende Vorteile:

- Die mehrfache Programmierung allgemeiner Funktionen als Hilfsklassen in den verschiedenen Anwenderbausteinen wird vermieden.
- Mit der Verwendung des jeweils aktuellen Basistoolsets ist sichergestellt, dass sämtliche allgemeinen Funktionen zur Verfügung stehen, die Versionskontrolle wird einfacher.
- Da wiederkehrende Funktionen nicht in jedem Objekt implementiert werden müssen, verringert sich die Größe der Objekte bzw. Modelle.
- Mit Hilfe des vorgefertigten Basistoolsets wird die Modellerstellung erleichtert.

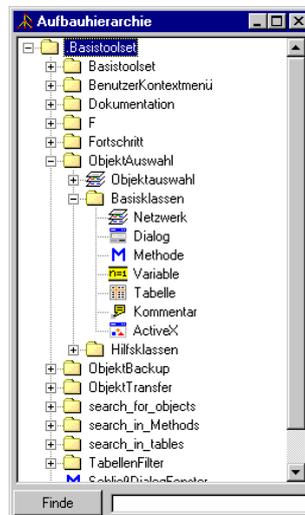


Abbildung 10 - Basistoolset: Aufbauhierarchie

Andererseits können die Bausteine des Toolsets nur in Modellen funktionieren, in denen das Basistoolset vorhanden ist.

Das Basistoolset ist in einem gleichnamigen Ordner in der Klassenbibliothek abgelegt (Abbildung 10). Es enthält u.a. einen Ordner, der ebenfalls den Namen Basistoolset trägt. In ihm sind eine zusammenfassende Dokumentation des gesamten Basistoolsets und ein Netzwerk mit dem Namen „autoexecMeth“ untergebracht.

AutoexecMeth

Das Netzwerk *.Basistoolset.Basistoolset.AutoexecMeth* wurde eingeführt, um Methoden, die nach dem Laden einer Simulationsmodelldatei automatisch ausgeführt werden sollen, koordiniert aufzurufen. Normalerweise werden in eM-Plant alle Methoden in der Klassenbibliothek, die den Namen „autoexec“ tragen, ohnehin automatisch ausgeführt, jedoch kann die Aufrufreihenfolge bei mehreren autoexec-Methoden in einem Modell nicht im voraus festgelegt werden.

Die Methode *.Basistoolset.Basistoolset.Autoexec* betrachtet eine Tabelle, in der - nach Priorität sortiert - Methoden eingetragen sind, die in dieser Phase aufgerufen werden sollen. Alle auf „aktiv“ gesetzten Methoden werden ausgeführt. Mit der Priorisierung lässt sich somit die Reihenfolge der Methodenaufrufe erzwingen.

Die weiteren Elemente des Basistoolsets befinden sich in den übrigen Ordnern. Diese Objekte wiederum sind gemäß Gestaltungsrichtlinie aus Basis- und Hilfsklassen aufgebaut. Die einzelnen Objekte des Basistoolsets werden im folgenden beschrieben.

2.2.2.1 Dokumentation



Die in eM-Plant erstellten Toolsetbausteine werden mit Hilfe dieses Tools dokumentiert und verwaltet. Neben der Möglichkeit zum Erstellen von Dokumentationen, die mit dem Baustein als eM-Plant-Objektdatei abgespeichert werden können, werden u.a. auch eine Schnittstelle zur Datenbank „Bausteinverwaltung“ und die Generierung von HTML-Dokumentationen von Methoden zur Verfügung gestellt. Nachfolgend wird anhand eines Beispiels der Einsatz der Bausteindokumentation beschrieben.

Anlegen einer neuen Dokumentation

Anhand Abbildung 11 kann nachvollzogen werden, wie für ein Objekt eine Dokumentation angelegt wird. Zu dokumentieren sei der Baustein „Dummy“ mit der Fachklasse *.Dummy.Dummy* (Abbildung 11 - 1.). Durch Doppelklick auf das Symbol des Netzwerks *.Basis-toolset.Dokumentation.Dokumentation* wird das GUI des Dokumentationsbausteins aktiviert (Abbildung 11 - 2.). Um einen Simulationsbaustein erstmalig zu dokumentieren, oder um die Dokumentation eines Bausteins zu editieren, wird der obere „Select“-Button des Dialogs (Abbildung 11 - 3.) betätigt. Im dann erscheinenden ActiveX-Control wird das Netzwerk ausgewählt, in dem die Dokumentation gespeichert werden soll, hier ist dies das Netzwerk *.Dummy.Dummy*. Der Pfad des Netzwerks wird dann im Dialog angezeigt.

Im gezeigten Fall wurde ein noch nicht dokumentierter Baustein ausgewählt. Damit die Dokumentation am Baustein selbst gespeichert werden kann, müssen nun einige Elemente (Informationsflussbausteine), die im Dokumentationsbaustein gespeichert sind, in den Dummy-Baustein transferiert werden (s.a. 2.2.2.6). Dazu wird der „Create“-Button im GUI betätigt (Abbildung 11 - 4.). Die benötigten Elemente werden in einem Netzwerk mit dem Namen „Dokumentation“ abgelegt, das in dem zu dokumentierenden Baustein erzeugt wird. Danach kann der Baustein dokumentiert werden. Später kann die Dokumentation des Bausteins jederzeit mit einem Doppelklick auf das Symbol des Netzwerks wieder editiert werden.

Dokumentation editieren

Die Dokumentation eines Bausteins ist in zwei Bereiche gegliedert, die vom GUI aus editiert werden.

Im Abschnitt „Allgemeine Dokumentation“ (s. Abbildung 12 - 1.) werden in strukturierter Form allgemeine Informationen zum Baustein gespeichert. Der obere Teil wird i.d.R. einmalig beim Anlegen des neuen Objekts ausgefüllt und enthält grundlegende Informationen, die in Tabelle 2 beschrieben sind. Der untere Teil wird fortlaufend ergänzt, um einen Überblick über die an späteren Objektversionen vorgenommenen Änderungen zu geben (Tabelle 3).

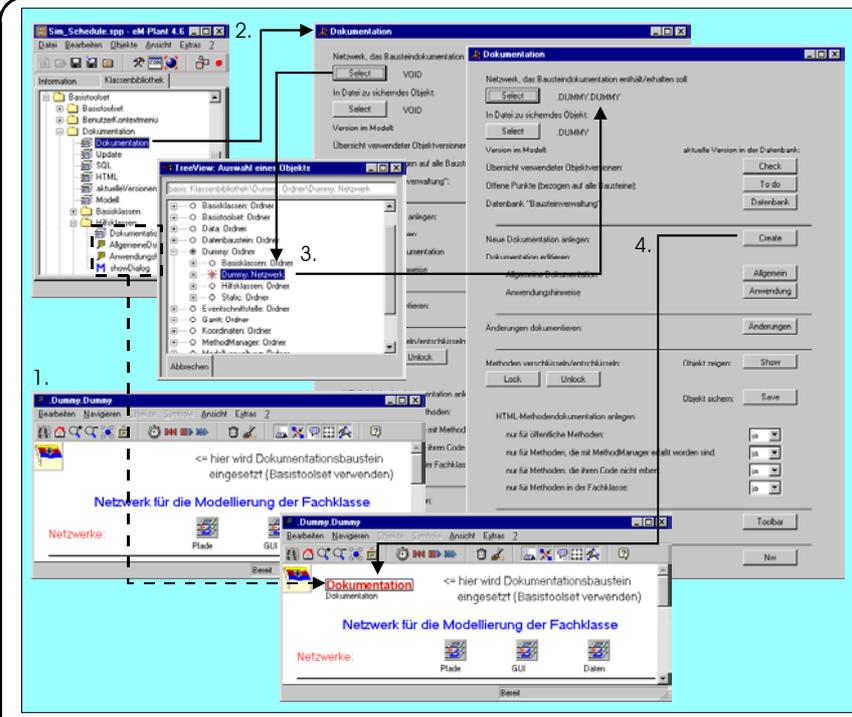
Eine ausführliche Beschreibung des Objekts in Textform kann im Abschnitt „Anwendungshinweise“ gespeichert werden (Abbildung 12 - 2.).

Baustein in Objektdatei speichern

Mit dem Button „Save“ wird das Speichern des Bausteins als Objektdatei ausgelöst (Abbildung 12 - 3.). Als Speicherort für alle Toolsetbausteine wurde ein Dateiordner eingerichtet. Gemäß Konvention ist dort für jeden Baustein manuell ein Unterordner einzurichten, der den Namen des Bausteins trägt, und in dem die Objektdateien zumindest der aktuellen Versionen des Bausteins zu speichern sind. Der Name der Datei soll aus dem Namen des

Bausteins und einer Kombination aus Versions- und Unterversionsnummer zusammengesetzt sein. Für den im Beispiel behandelten Baustein würde der Dateiname lauten:

... \Simple++ Objekte \Dummy \Dummy001000.obj ⇒ Baustein „Dummy“, Version 1, Unterversion 0



Dokumentation vorbereiten

1. zu dokumentierender Baustein
2. Dokumentationsbaustein aktivieren
3. zu dokumentierenden Baustein wählen
4. Elemente zum dezentralen Speichern der Dokumentation in neuen Baustein transferieren

Basistoolset - Dokumentation I



Abbildung 11 - Dokumentationsbaustein: Anlegen einer neuen Dokumentation

„Objekt“:	Name des Netzwerks, in das die Dokumentation eingesetzt wurde; Eintrag erfolgt automatisch
„Objekt-Version“:	Versionsnummer, bestehend aus Versions- und Unterversionsnummer. Bei Erstdokumentation wird „1.0“ eingetragen, beim Erfassen neuer Bausteinversionen wird die aktuelle Versionsnummer eingetragen; Eintrag erfolgt automatisch
„Simple-Version“:	eM-Plant-Versionsnummer zur Zeit der Erstdokumentation, bestehend aus Versions-, Unterversions- und Revisionsnummer; Eintrag erfolgt automatisch
„Kurze Beschreibung“:	Knappe Beschreibung der Objektfunktionalität; Eintrag manuell
„Autor“:	Name des Objekterstellers. bei Erstdokumentation wird Rechnername eingetragen, manuelle Änderung möglich.
„Zuständig f. Änderungen“:	Verantwortlicher für die Objektpflege; bei Erstdokumentation wird Rechnername eingetragen, manuelle Änderung möglich.
„Datum Implementations“:	Datum der Erstdokumentation eingetragen; Eintrag erfolgt automatisch
„Abgelegt auf“:	Dateipfad der Objektdatei; Eintrag erfolgt automatisch

Tabelle 2 - Dokumentationsbaustein: allgemeine bausteinbezogene Informationen

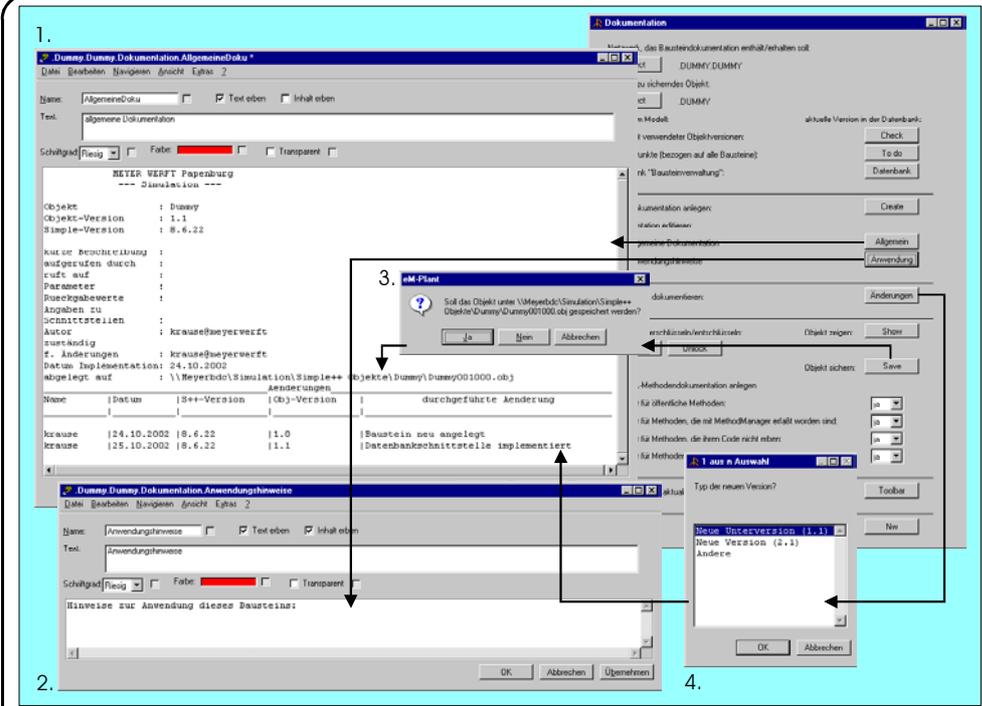
„Name“:	Name des Änderers; es wird automatisch der Rechnername eingetragen; manuelle Änderung möglich
„Datum“:	Datum der Änderung; Eintrag erfolgt automatisch
„S++Version“:	eM-Plant-Version zum Zeitpunkt der Änderung
„Obj-Version“:	Objektversionsnummer; Eintragung im Dialog mit dem Anwender
„Durchgeführte Änderung“:	knappe Beschreibung der Änderung; manuell auszufüllen

Tabelle 3 - Dokumentationsbaustein: allgemeine versionenbezogene Informationen

Ein entsprechender Name wird generiert und muss vor dem Sichern des Bausteins vom Anwender bestätigt oder verworfen werden. Der Baustein wird dann gespeichert. Anschließend wird ein Update der Bausteindatenbank vorgenommen und auf Wunsch eine Methodendokumentation im HTML-Format angefertigt.

Versionspflege

Wurde der Baustein weiterentwickelt, so können die vorgenommenen Erweiterungen bzw. Änderungen anschließend ebenfalls dokumentiert und eine aktualisierte Version abgespeichert werden. Hierfür ist im GUI der Button „Änderungen“ vorgesehen (Abbildung 12 - 4). Nach dessen Betätigung erscheint eine Eingabeaufforderung bezüglich der Art der Änderung (neue Version / Unterversion / andere), und der Baustein „Allgemeine Dokumentation“ wird geöffnet, damit dort die Dokumentation aktualisiert werden kann. Abschließend ist der Baustein in der oben beschriebenen Weise zu speichern.



Dokumentation bearbeiten

1. Allgemeine Dokumentation
2. Anwendungshinweise
3. Baustein sichern
4. Änderungen dokumentieren

Basistoolset - Dokumentation II



Abbildung 12 - Dokumentationsbaustein: Dokumentation bearbeiten

Versionskontrolle

Ein Simulationsmodell enthält i.d.R. viele Simulationsbausteine, die u.U. nach der Erstellung des Modells weiterentwickelt worden sein können. Mit Hilfe der Versionskontrolle kann schnell ermittelt werden, welche Bausteine in einem Modell vorhanden sind, und ob die jeweils aktuelle Version verwendet wird. Diese Kontrolle wird mit dem „Check“-Button des GUIs gestartet. Alle im Modell enthaltenen Bausteine werden ausgewertet. Zudem wird geprüft, ob der Baustein in der Datenbank erfasst worden ist. Resultat der Auswertung ist eine Tabelle, in der die im Modell enthaltenen Bausteine und ihre (Unter-)versionsnummer aufgeführt sind, und die Angabe, ob in der Datenbank eine höhere (=aktuellere) Version erfasst wurde (Abbildung 13 - 1.).

Beim Editieren der Dokumentation eines einzelnen Bausteins wird außerdem im GUI die Versionsnummer des aktuell bearbeiteten Bausteins und die der höchsten in der Datenbank gefundenen Version des selben Bausteins angezeigt (Abbildung 13 - 2.).

HTML - Methodendokumentation

Im Zuge des Sicherns des Objekts kann eine Dokumentation der Methoden im HTML-Format erstellt werden (Abbildung 13 - 4). Hierzu wird der MethodManager von Tecnomatix benötigt. Die Methoden, die in diesem Zusammenhang benötigt werden, befinden sich im Netzwerk .Basistoolset.Dokumentation.HTML.

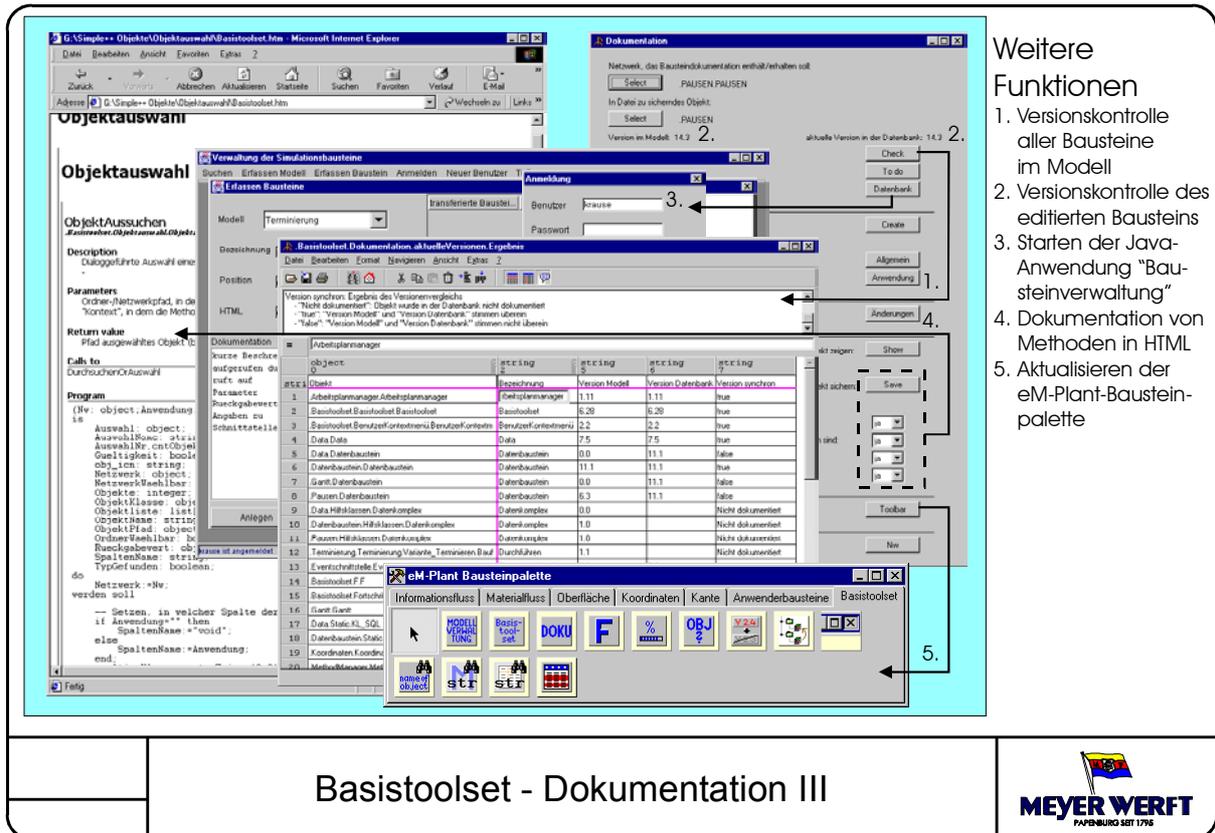
Unter Umständen sollen nicht alle Methoden des Bausteins in der Dokumentation aufgeführt werden. Im Dialog kann die Auswahl wie folgt eingeschränkt werden:

- „nur für öffentliche Methoden“: es werden nur Methoden dokumentiert, die als aktuelles Icon das Symbol „User“ (grüner Buchstabe „M“) verwenden 
- „nur für Methoden, die mit MethodManager erfasst worden sind“: HTML-Dokumentation erfolgt nur, wenn der Methodenkopf mit dem MethodManager erstellt worden ist.
- „nur für Methoden, die ihren Code nicht erben“: keine HTML-Dokumentation bei Methoden, deren Programmcode von einer anderen Methode geerbt wird.
- „nur für Methoden in der Fachklasse“: keine HTML-Dokumentation bei Methoden, die Hilfsfunktionen enthalten, wie z.B. die Netzwerke „GUI“ oder „SQL“.

Bausteinpalette

eM-Plant stellt eine Toolbar zur Verfügung, auf der Verknüpfungen zu häufig benötigten Bausteinen im Modell visualisiert werden können. So kann auf dieser Ebene beispielsweise dem Simulationsanwender eine Auswahl aller relevanten Bausteine zur Verfügung gestellt werden, um damit Modelle zu erstellen, ohne dafür die eigentliche Bausteinbibliothek zu zeigen.

Um die Funktionalität der Bausteinpalette effektiver zu nutzen, wurde eine Funktion integriert, die mit dem Button „Toolbar“ aktiviert wird. Das aktuelle Modell wird nach string-Variablen mit dem Namen „Toolbar“ durchsucht. Alle Bausteine, die diese Variable enthalten, werden dann auf der Bausteinpalette auf einer Karteikarte des mit dem Wert der Variablen bezeichneten Namens abgebildet. So wird das Modellieren bzw. die Arbeit mit dem Modell erleichtert (Abbildung 13 - 5.).



Basistoolset - Dokumentation III

Abbildung 13 - Dokumentationsbaustein: weitere Funktionen

Datenbankanbindung

Bei jedem Sichern eines Bausteins wird im Hintergrund ein Update der Datenbank „Baustein-dokumentation“ durchgeführt. Es werden Informationen zum Baustein, seinen Versionen und zu den enthaltenen Methoden ausgewertet:

- **Baustein:** Die im oberen Teil der Allgemeinen Tabellendokumentation gemachten Angaben werden ausgewertet und in der Datenbank in folgende Felder der Tabelle „Baustein“ geschrieben:
 - *Dokumentation* (Kurzbeschreibung des Objekts)
 - *Position* (Pfad der Objektdatei)
 - *HTMLDoku* (Pfad der HTML-Dokumentation, sofern erstellt)
(Schlüssel: Bezeichnung des Bausteins)
- **Versionen:** Die im unteren Teil der Allgemeinen Tabellendokumentation gemachten Angaben werden ausgewertet und in der Datenbank in folgende Felder der Tabelle „Versionen“ geschrieben:
 - *Beginn* (Datum der Änderung)
 - *Ende* (Datum der Änderung)
 - *Aenderer*
 - *SimpleVNr*
 - *Dokumentation* (Kurzbeschreibung der Änderung)
(Schlüssel: Bausteinbez, Versionsnr, Uversionsnr)

- Methoden: Es werden diejenigen öffentlichen und mit dem MethodManager dokumentierten Methoden ausgewertet, die sich im selben Netzwerk befinden, in dem das Dokumentationsnetzwerk eingesetzt ist. In die Tabelle „Methoden“ der Datenbank wird in folgendes Feld der Tabelle „Methoden“ geschrieben:
 - *Dokumentation* (Methodenkopf, erstellt mit Tecnomatix-MethodManager) (Schlüssel: Baustein, Bezeichnung der Methode)

Die im Dokumentationsnetzwerk gespeicherten Angaben werden ausgewertet und in die drei Tabellen tab_Baustein, tab_Versionen und tab_Methoden geschrieben. Die Informationen werden anschließend in die gleichnamigen Tabellen der Datenbank gespeichert.

Mit dem „Datenbank“-Button kann die „Bausteinverwaltung“ (vgl. 2.5.4.2) aktiviert werden, um ggf. die Dokumentation zu ergänzen, z.B. Eingabe von Schlagwörtern.

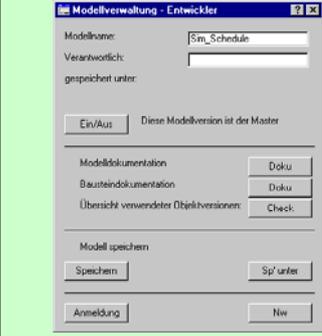
2.2.2.2 Modellverwaltung



In diesem Baustein sind modellbezogene Funktionen modelliert. Dazu gehören die Dokumentation und Versionsverwaltung von Simulationsmodellen, ein Simulationscockpit als allgemeine Anwenderoberfläche, unterschiedliche Darstellungsformen für Anwender und Entwickler und das Schreiben von Logdaten in die Datenbank „SimAnmeldung“.

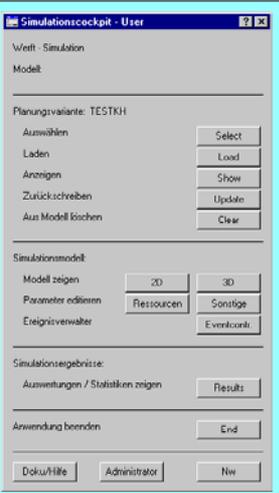
C. GUI "Modellverwaltung" (Entwicklersicht)

1. Allgemeine Informationen zum Modell
2. Dokumentations- und Versionspflege
3. Modell speichern
4. Hilfsfunktionen



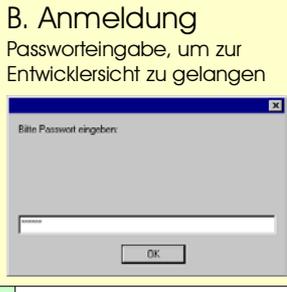
A. GUI "Simulationscockpit" (Anwendersicht)

1. Name des Modells
2. Zugriff auf Eingangsdaten (Planungsvarianten)
3. Zugriff auf Simulationsmodell (Parametrierung, Visualisierung, Experiment durchführen)
4. Auswertung von Experimenten
5. eM-Plant beenden
6. Hilfsfunktionen



B. Anmeldung

Passworteingabe, um zur Entwicklersicht zu gelangen



Modellverwaltung: Dialoge und Funktionen



Abbildung 14 - Modellverwaltung: Dialoge und Funktionen

- GUI „Simulationscockpit“: allgemeine Oberfläche für Anwender von Simulationsmodellen, das die wesentlichen Modellfunktionen zur Verfügung stellt, modellspezifische Besonderheiten werden in speziell anzupassenden Dialogen abgehandelt (Abbildung 14-A).

- GUI „Modellverwaltung“: allgemeine Oberfläche für Entwickler von Simulationsmodellen, das weitergehende Funktionen zur Modellpflege anbietet (Abbildung 14-C).
- Konfiguration und Verwaltung der Modellstati: Anwendern und Entwicklern wird ein unterschiedliches Aussehen und Verhalten des Modells geboten. Folgende Eigenschaften können beeinflusst werden:
 - Sichtbarkeit von Dialogen und bestimmten Dialogelementen sowie der Standardmenüs in den Netzwerken
 - Sichtbarkeit der Klassenbibliothek und der Bausteinpalette
 - Schutz von Dialogen gegen unbeabsichtigtes Schließen
 - Sperren von Informationsflussbausteinen gegen unbeabsichtigte Änderungen
 - Verschlüsselung der Methodentexte

Nach dem Öffnen der Modelldatei wird das Modell dem abgespeicherten „Init-Status“ angepasst. Nach Eingabe eines Passwortes kann der Status von „Anwender“ zu „Entwickler“ geändert werden (Abbildung 14-B).

- Schreiben von Logdaten in eine Datenbank: Anhand der Daten kann nachvollzogen, wann die jeweiligen Modelle von den Mitarbeitern verwendet wurde.

2.2.2.3 Fortschritt



Werden bei Methodenaufrufen oder Simulationen zeitaufwändige Routinen durchlaufen, so ist dieser Baustein nützlich, um den Anwender währenddessen über den Fortschritt der Abarbeitung zu informieren. Das Fortschritt-Objekt wird mit dem Aufruf der Methode `.Basistoolset.Fortschritt.Fortschritt.zeigeFortschritt` aktiviert. Zu übergeben ist die Referenz auf eine zu durchlaufende Zählvariable, der Zielwert dieser Variable, bei der die Bearbeitung der Schleife endet, und ein kurzer Text, der die Aktivität beschreibt (s. Beispiel).

Beim Einsatz des Fortschritt-Objekts ist folgendes zu beachten:

- Bei der Zählvariable muss es sich um eine globale (beobachtbare) Variable handeln, da ihr Wert mit einer `waituntil`-Anweisung überwacht wird.
- Der Zielwert der Variable, bis zu dem die Schleife zu durchlaufen ist, muss im voraus bekannt sein und an die Methode `zeigeFortschritt` übergeben werden, damit der prozentuale Abarbeitungsrad errechnet werden kann.
- Der Startwert der Zählvariable muss vor Aktivierung der Methode `zeigeFortschritt` gesetzt werden
- Das Fortschrittobjekt funktioniert nicht, wenn rückwärts gezählt wird.
- Da mit einer `waituntil`-Anweisung der Wert der Zählvariable kontrolliert wird, kann die Performance beeinträchtigt werden.
- Bei Aktivierung des Objekts werden temporäre Netzwerke erzeugt, die bei vorzeitigem Abbruch der Schleife nicht vernichtet werden. Um diese Netzwerke zu vernichten, sollte regelmäßig die `RESET`-Methode des Fortschritt-Objekts aktiviert werden (eine `autoexec`-Methode sorgt für regelmäßiges Zurücksetzen beim Laden eines Modells, das das Fortschritt-Objekt enthält).



Abbildung 15 - Fortschritt: Beispiel Fortschrittmeldung

Beispiel:

```
is
M_Fortschritt: object; --öffentliche Methode des Fortschritt-Objekts
Zaehler: object;      --Referenz auf Zählvariable
Zielwert: integer;    --Zielwert der Schleife
do
M_Fortschritt:=ref(.Basistoolset.Fortschritt.Fortschritt.zeigeFortschritt);
Zaehler:=ref(current.Counter);
Zielwert:=Lieferliste.yDim;

from Zaehler.wert:=1;      -- Zaehler ist ein NwData vom Typ integer oder real
M_Fortschritt.ausführen(Zaehler,Zielwert,"Lieferliste wird angelegt.");
until Zaehler.wert>Zielwert
loop
LieferlisteAnlege(Zaehler.wert);
Zaehler.wert:=Zaehler.wert+1;
end;
end;
```

2.2.2.4 Objektauswahl



Dieser Baustein stellt u.a. Methoden zur dialoggeführten Auswahl von Simulationsobjekten zur Verfügung. Die Auswahl erfolgt nach Möglichkeit mit Hilfe der ActiveX-Komponenten „TreeView“ und „ListView“, andernfalls mit eM-Plant-Standardelementen. Weitere Methoden dienen zur Objektsuche in bestimmten Bereichen eines Simulationsmodells nach vorzugebenden Kriterien.

Öffentliche Methoden

ObjektAusTree, ObjektAussuchen

Mit diesen beiden Methoden wird der Anwender aufgefordert, genau ein Objekt auszuwählen. Die Methode erwartet in beiden Fällen zwei Parameter:

- einen Objektpfad (Ordner, Netzwerk oder das Schlüsselwort „Basis“ für die gesamte Klassenbibliothek). Dieser Pfad gibt an, aus welchem Teil der Klassenbibliothek ein Objekt ausgewählt werden soll. Nur Elemente innerhalb des angegebenen Objekts können ausgewählt werden.
- einen String, der auf den Kontext hinweist, in dem die Objektauswahl stattfinden soll. Im Baustein Objektauswahl ist in einer Tabelle hinterlegt, welche Basisklassenbausteine je nach angegebenem Kontext wählbar sind. Z.B. können vom Pausenbaustein nur Materialflussbausteine und Netzwerke als Ressourcen nach einem Schichtkalender getaktet werden. Werden die o.g. Methoden mit dem Kontext „Pause“ aufgerufen, so wird die Möglichkeit, ein Objekt anderen Typs auszuwählen, von vornherein unterdrückt.

Beim Aufruf der Methode „ObjektAusTree“ werden die auswählbaren Objekte mit Hilfe einer ActiveX-Steuerung in einem TreeView angezeigt, dabei werden der Objektname, ggf. das Etikett und der deutsche Name der entsprechenden Basisklasse aufgeführt (ähnlich Abbildung 16 - links). Falls die benötigten ocx-Dateien fehlen, wird die alternative Methode „Objekt-

Aussuchen“ aufgerufen, die die Objekte in einer promptList darstellt (Abbildung 17). Rückgabewert ist der Pfad des ausgewählten Objekts.

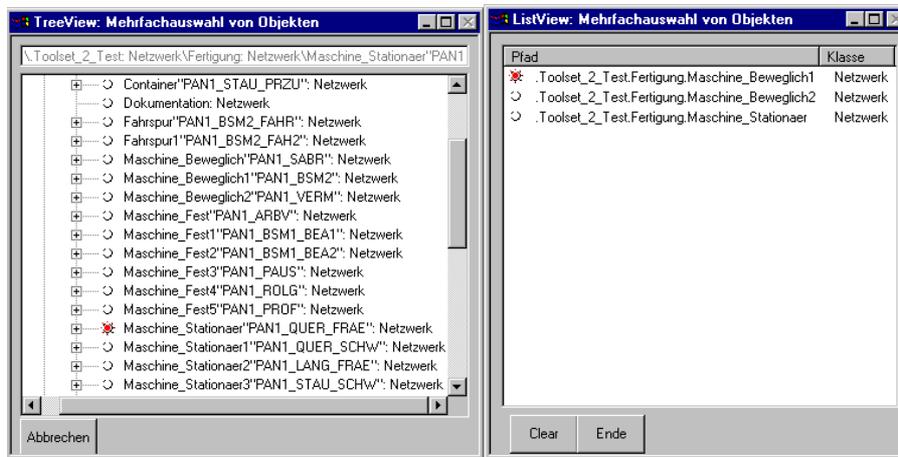


Abbildung 16 - Objektauswahl: Mehrfachauswahl mit TreeView und ListView

ObjekteAusTree, ObjekteAussuchen

Diese Methoden bieten die Möglichkeit, mehrere Objekte auf einmal auszuwählen. Die Parameter sind die gleichen wie bei „ObjektAusTree“ und „ObjektAussuchen“, zurückgegeben wird eine Liste von Objektpfaden. Beim Aufruf der Methode „ObjekteAusTree“ werden, sofern die benötigten ocx-Dateien vorliegen, die auswählbaren und die ausgewählten Objekte mit den ActiveX-Komponenten „TreeView“ und „ListView“ angezeigt (Abbildung 16), andernfalls mit dem eM-Plant-Standardelement promptListN.

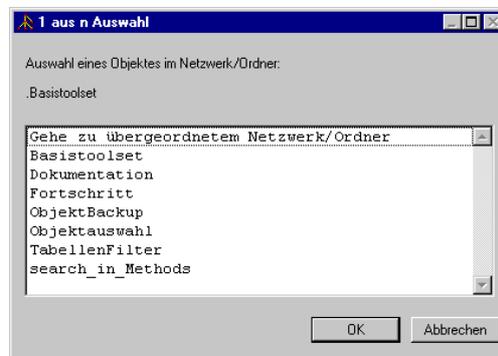


Abbildung 17 - Objektauswahl: Eingabeaufforderung mit promptList

Search_in

Auch dieser Methode wird als erster Parameter ein Objektpfad übergeben, der angibt, in welchem Teil des Modells gesucht werden soll. Der zweite Parameter bezeichnet einen Objekttyp (Attribut „internalClassName“). Die Methode liefert eine Tabelle mit zwei Spalten. In der ersten stehen die Pfade aller innerhalb des angegebenen Bausteins gefundenen Objekte, für die der angegebene internalClassName mit diesem Wert übereinstimmt.

Get_valid_objects

Die Methode benötigt die gleichen Parameter wie ObjektAusTree, ObjektAussuchen, ObjekteAusTree und ObjekteAussuchen. Sie wertet den angegebenen Kontext aus und ruft iterativ für jeden in diesem Kontext auswählbaren Bausteintyp die Methode „search_in“ auf. Die gefundenen Bausteine werden in eine gemeinsame Tabelle geschrieben. Der Rückgabewert ist

eine Tabelle aller Objekte innerhalb eines angegebenen Teils des Simulationsmodells, die in einem bestimmten Kontext auswählbar sind.

ObjekteAktion

An diese Methode ist eine Liste von Objekten zu übergeben. Ein zweiter Parameter enthält eine in SimTalk formulierte Anweisungssequenz. Die Methode ObjekteAktion wandelt diese in einen vollständigen Methodentext um. Die Methode erwartet einen Parameter vom Typ object, sein Name ist „obj“. Die Objektliste wird dann durchlaufen und der Methodentext je einmal ausgeführt, mit dem jeweiligen Objekt aus der Liste als Parameter. Bei Syntax- oder sonstigen Fehlern wird die Ausführung der Methode abgebrochen, es erfolgt keine Fehlermeldung oder Debugging. In die Konsole wird eine Meldung geschrieben, bei wie vielen Objekten die Ausführung der Methode gelang.

ObjekteFilter

Diese Methode ermöglicht das Auffinden von Objekten nach vorgebbaren Suchkriterien. Die Methode erwartet als Parameter den Suchraum (Netzwerk, Ordner oder das Schlüsselwort „basis“, falls in der gesamten Klassenbibliothek gesucht werden soll) und eine Schlüsseltabelle, in der die Filterkriterien definiert sind. Sie enthält, in SimTalk formuliert, je Zeile eine Bedingung. Verweise auf das Objekt werden mit dem Schlüsselwort „obj“ ausgedrückt. Mehrere Bedingungen in einer Zeile gelten als UND-Verknüpfung, die in unterschiedlichen Zeilen eingetragenen Bedingungen sind miteinander ODER-verknüpft. Zurückgegeben wird eine Tabelle, in der die Pfade der gefundenen Objekte und der jeweilige „internalClassName“ aufgeführt sind. Abbildung 18 zeigt ein Beispiel für den Einsatz der Methode: Links ist eine ausgefüllte Schlüsseltabelle zu sehen, rechts die Tabelle mit den gefundenen Objekten. Gesucht wurden Objekte, die den Namen „Kommentar1111“ tragen (erste Zeile) *oder* dokumentierte Bausteinklassen sind (zweite Zeile).

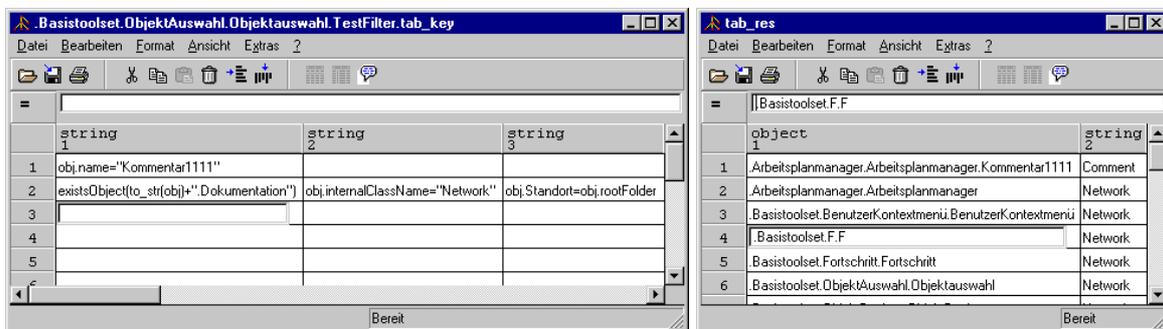


Abbildung 18 - Objektauswahl: Beispiel Methode ObjekteFilter

2.2.2.5 ObjektBackup

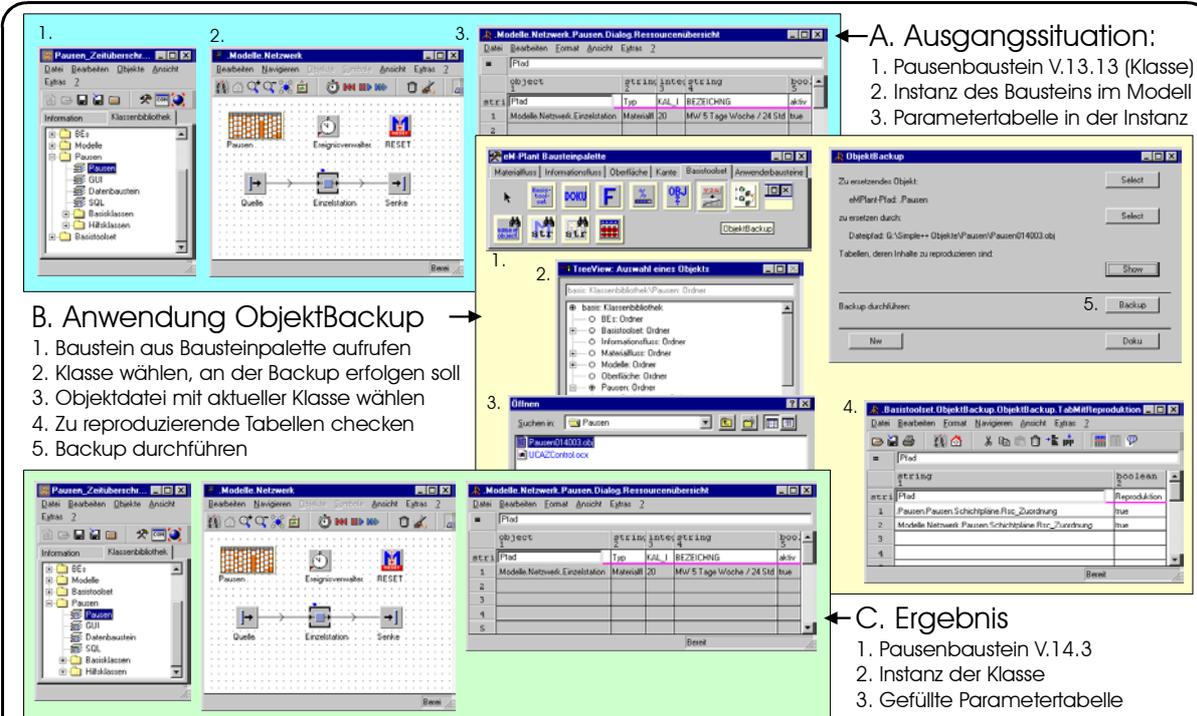


Der Baustein „ObjektBackup“ erleichtert den Austausch von Anwenderbausteinen, die in einem Modell eingesetzt sind und z.B. durch eine aktualisierte Version des Bausteins ersetzt werden sollen. Daten, die in Instanzen des zu ersetzenden Bausteins gespeichert sind und bei der Zusammenführung der Klassen normalerweise verloren gingen, können bei Bedarf anschließend reproduziert werden.

Anwendungsbeispiel

In Abbildung 19 ist die Vorgehensweise bei der Anwendung dieser Funktionalität für einen typischen Fall skizziert. Der Pausenbaustein (2.2.6.1) ist in einem Modell zunächst in der Version 13.13 eingesetzt. Eine Instanz der Fachklasse befindet sich in einem Netzwerk und enthält eine Tabelle, in der modellspezifische Parameter eingetragen wurden, in diesem Fall

Pfade zu Ressourcen, deren Pausenzustand der Baustein in der Simulation steuern soll (Abbildung 19 - A).



A. Ausgangssituation:

1. Pausenbaustein V.13.13 (Klasse)
2. Instanz des Bausteins im Modell
3. Parametertabelle in der Instanz

B. Anwendung ObjektBackup

1. Baustein aus Bausteinpalette aufrufen
2. Klasse wählen, an der Backup erfolgen soll
3. Objektdatei mit aktueller Klasse wählen
4. Zu reproduzierende Tabellen checken
5. Backup durchführen

C. Ergebnis

1. Pausenbaustein V.14.3
2. Instanz der Klasse
3. Gefüllte Parametertabelle

Basistoolset - ObjektBackup



Abbildung 19 - ObjektBackup: Anwendungsbeispiel

Zwischenzeitlich wurde der Pausenbaustein weiterentwickelt und soll nun durch die aktuelle Version 14.3 ersetzt werden. Ohne die ObjektBackup-Funktionalität wären nun manuell Version 13.13 aus dem Modell zu löschen und Variante V.14.3 als Objektdatei in die Klassenbibliothek zu laden gewesen; danach hätten alle Instanzen erneut modelliert und parametrieren werden müssen - eine aufwändige und fehleranfällige Methode, die mit dem ObjektBackup-Baustein weitgehend automatisiert ist. Noch vor dem Entfernen der alten Bausteinversion werden alle Instanzen des Bausteins ermittelt und die für die Reproduktion nach dem Laden der neuen Version benötigten Informationen gesichert. Ferner wird das gesamte Modell nach Tabelleneinträgen durchsucht, die Verweise auf Elemente des alten Pausenbausteins bzw. dessen Instanzen enthalten. Nach dem Austauschen des Bausteins müssen diese Verweise ebenfalls reproduziert werden.

Der alte Pausenbaustein wird dann aus dem Modell entfernt, der neue wird - aus einer Objektdatei - in das Modell geladen. Anschließend wird dieser Baustein genau dort instanziiert, wo es zuvor der alte war, die Parametrierung wird dem Ausgangszustand entsprechend wiederhergestellt und die Verweise auf die Instanzen des Pausenbausteins werden in den übrigen Objekten reproduziert (Abbildung 19 - B und C).

2.2.2.6 ObjektTransfer



Dieser Baustein ermöglicht es, Teile eines Bausteins in einen anderen Baustein zu verschieben und dabei die Vererbungsbeziehungen so zu manipulieren, dass die verschobenen Elemente anschließend ausschließlich aus Basis- und Hilfsklassen ihres

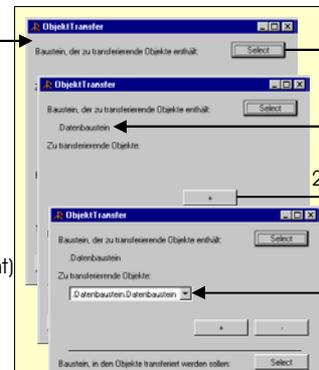
neuen Standortbausteins aufgebaut sind. Im Unterschied zum Baustein „ObjektBackup“ (2.2.2.5) findet dieser Baustein Anwendung, wenn z.B. einem bestehenden Anwenderbaustein eine neue Komponente hinzugefügt werden, im übrigen der Baustein aber unverändert bleiben soll.

Anwendungsbeispiel

In Abbildung 20 ist die Vorgehensweise bei der Anwendung dieser Funktionalität für einen typischen Fall skizziert. In einem Modell befinden sich die Objekte „Pausenbaustein“, „Datenbaustein“ und das Basistool „ObjektTransfer“. Der in diesem Beispiel verwendete Pausenbaustein enthält keine Schnittstelle zu der Datenbank, in der Schichtkalender beschrieben sind. Der Datenbaustein bietet die Funktionalitäten, die für solch eine Schnittstelle benötigt werden (Abbildung 20 - A).

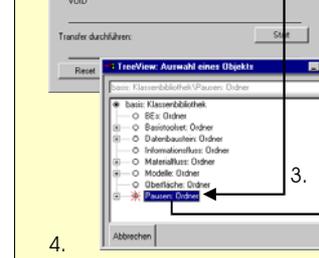
A. Ausgangssituation:

1. Pausenbaustein
2. Datenbaustein
3. GUI „ObjektTransfer“ aktivieren



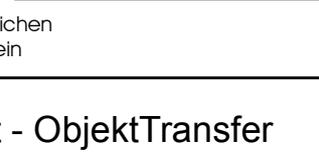
B. Anwendung ObjektTransfer

1. Objekt „Datenbaustein“ wählen (enthält das zu transferierende Element)
2. Netzwerk „Datenbaustein.“ wählen (zu transferierendes Element)
3. Pausenbaustein wählen (Zielbaustein des Transfers)
4. Transfer durchführen



C. Ergebnis

1. Pausenbaustein mit eigenem Datenbaustein
2. Keine Vererbung zwischen Elementen des ursprünglichen Datenbausteins und dessen Kopie im Pausenbaustein



Basistoolset - ObjektTransfer



Abbildung 20 - ObjektTransfer

Um den Pausenbaustein mit der beschriebenen Schnittstelle auszustatten, wäre es z.B. möglich, die Fachklasse des Datenbausteins in den Hauptordner des Pausenbausteins zu instanzieren. Dann enthielte aber der Pausenbaustein Elemente, deren Klassen sich außerhalb des Ordners „Pausen“ befinden und das Prinzip, alle Bausteine des Simulationstoolsets als in sich „abgeschlossene“ Ordner zu halten, wäre verletzt.

Alternativ könnte der gesamte Ordner „Datenbaustein“ in den Ordner „Pausen“ verschoben werden. Das beschriebene Problem wäre damit zwar gelöst, aber die Basis- und diverse Hilfsklassen kämen im Pausenbaustein mehrmals vor, was zu unnötig großen Modellen führt. eM-Plant bietet die Möglichkeit, Objekte gleichen Aufbaus manuell zusammenzuführen, allerdings ist diese Methode aufwändig und fehlerträchtig.

Im ObjektTransfer-Baustein ist dieser Ablauf weitgehend automatisiert. Das GUI des Bausteins kann sowohl mit Doppelklick auf das Symbol des Netzwerks *.Basistoolset.ObjektTransfer.ObjektTransfer* als auch über das entsprechende Symbol in der Toolbar aktiviert werden. Anschließend werden die Objekte ausgewählt, die in einen anderen Baustein transferiert werden sollen, sowie derjenige Baustein, in den diese Elemente hinein zu verschieben sind (Abbildung 20 - B). Nach dem Transfer ist der gewünschte Zustand (hier: Pausenbaustein mit eigener Datenbankschnittstelle) hergestellt (Abbildung 20 - C). Zu beachten ist, dass der Baustein, der die zu transferierenden Elemente aufnimmt, keinerlei Änderung erfährt.

2.2.2.7 Search_in_Methods



Der Search_in_Methods-Baustein bietet Funktionen zur Volltextsuche in eM-Plant-Methodenbausteinen mit optionaler Ersetzungsfunktion (Weiterentwicklung eines Tecnomatix-Produktes)

Das GUI des Bausteins (Abbildung 21 - links) kann sowohl mit Doppelklick auf das Symbol des Netzwerks *.Basistoolset.Search_In_Methods.Search_In_Methods* als auch auf das entsprechende Symbol auf der Bausteinpalette (rechte Maustaste) aktiviert werden. Dort bestehen dann folgende Auswahlmöglichkeiten:



Abbildung 21 - searchInMethods: Benutzeroberfläche und Ergebnistabelle

- Suchstring: Eingabe einer Zeichenkette, die in den Methoden gefunden werden soll
- Netzwerk: Eingabe des Suchraums. Eingetragen werden kann ein Netzwerk oder Ordner. Erfolgt keine Eingabe, so werden alle Methoden im gesamten Modell durchsucht, andernfalls die Methoden im angegebenen Element und in allen darunter liegenden Ordnern und Netzwerken.
- Groß-/Kleinschreibung: Wird „Nein“ gewählt, so werden Texte als übereinstimmend angesehen, wenn sie sich allenfalls in der Groß-/Kleinschreibung unterscheiden.
- Erkennung von Wörtern: Wird „Ja“ gewählt, so werden nur Übereinstimmungen als gültig angesehen, die im Methodentext als ganzes Wort vorkommen (Beispiel: gesucht wird „Pause“, in einer Methode steht das Wort „Pausenbaustein“ → keine gültige Übereinstimmungen, da „Pause“ als Bestandteil eines längeren Wortes verwendet wird.)
- Nur Klassen anzeigen: Wird „Ja“ gewählt, so werden nur Übereinstimmungen als gültig angesehen, die in Methoden gefunden wurden, die ihren Code nicht erben.
- Suche: Die Betätigung dieses Buttons startet die Suche nach Methoden, die Übereinstimmungen mit den zuvor eingegebenen Kriterien aufweisen. Das Ergebnis der Suche

wird in einer Tabelle (Abbildung 21 - rechts) ausgegeben, in der Verweise auf die gefundenen Methoden und die Methodenzeile aufgeführt sind.

- Suchen und Ersetzen / ersetzen durch: Über die reine Suchfunktion hinaus kann der Methodentext mit dieser Funktion auch verändert werden. An allen Fundstellen wird der Suchtext aus dem Code gelöscht und durch den im Eingabefeld „ersetzen durch“ eingetragenen Text ersetzt. Veränderungen werden nicht an Methoden durchgeführt, die ihren Text erben.

2.2.2.8 Search_for_objects



Search_for_objects stellt eine Suchfunktion für Objekte zur Verfügung. Suchkriterium ist dabei eine anzugebende Zeichenfolge, mit der die Namen der zu findenden Objekte übereinstimmen oder die in deren Namen enthalten sein sollen.

Die Benutzeroberfläche des Bausteins (Abbildung 22 - links) kann sowohl mit Doppelklick auf das Symbol des Netzwerks `.Basistoolset.search_for_objects.search_for_objects` als auch auf das entsprechende Symbol auf der Bausteinpalette (rechte Maustaste) aktiviert werden. Dort bestehen dann folgende Auswahlmöglichkeiten:

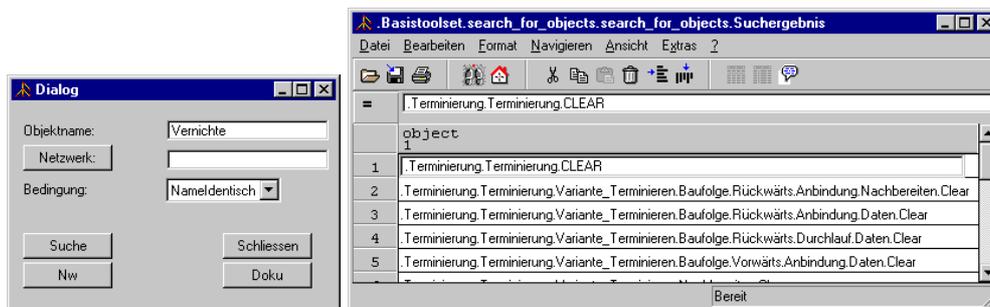


Abbildung 22 - search_for_objects: Benutzeroberfläche und Ergebnistabelle

- Objektname: Eingabe einer Zeichenkette, mit der Objektnamen verglichen werden sollen
- Netzwerk: Eingabe des Suchraums. Eingetragen werden kann ein Netzwerk oder Ordner. Erfolgt keine Eingabe, so werden alle Objekte im gesamten Modell durchsucht, andernfalls die Objekte im angegebenen Element und in allen darunter liegenden Ordnern und Netzwerken.
- Bedingung: Wird „NameIdentisch“ gewählt, so werden nur Objekte als gültige Übereinstimmungen angesehen, deren Name mit dem angegebenen String übereinstimmen, ansonsten („Teilstring“) alle Objekte, deren Name den angegebenen String enthalten.
- Suche: Die Betätigung dieses Buttons startet die Suche nach Objekten, die Übereinstimmungen mit den zuvor eingegebenen Kriterien aufweisen. Das Ergebnis der Suche wird in einer einspaltigen Tabelle (Abbildung 22 - rechts) ausgegeben, in der Verweise auf die gefundenen Objekte aufgeführt sind.

2.2.2.9 Search_in_tables

Search_in_tables stellt eine Suchfunktion für Einträge in Tabellen zur Verfügung. Suchkriterium ist dabei eine anzugebende Zeichenfolge. In Tabellenbausteinen wird dann nach Übereinstimmungen gesucht (bisher nur identische Übereinstimmungen).

Die Benutzeroberfläche des Bausteins (Abbildung 23 - links) kann sowohl mit Doppelklick auf das Symbol des Netzwerks *.Basistoolset.search_in_tables.search_in_tables* als auch auf das entsprechende Symbol in der Toolbar (rechte Maustaste) aktiviert werden. Dort bestehen dann folgende Auswahlmöglichkeiten:

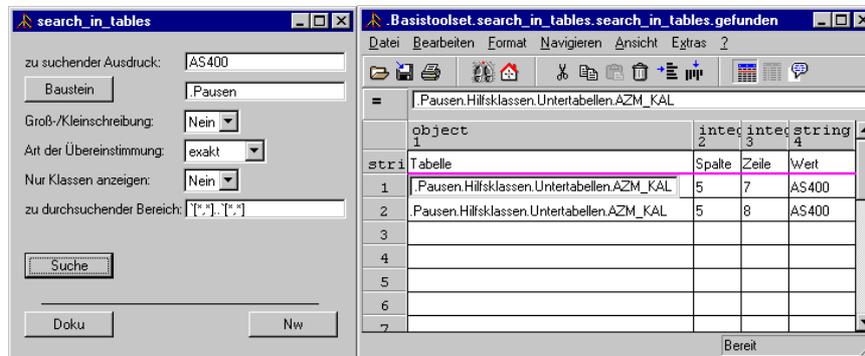


Abbildung 23 - search_in_tables: Benutzeroberfläche und Ergebnistabelle

- zu suchender Ausdruck: Zeichenkette, nach der gesucht werden soll
- Baustein: Eingabe des Suchraums oder einer einzelnen Tabelle, die durchsucht werden soll. Erfolgt keine Eingabe, so werden alle Tabellenbausteine im gesamten Modell durchsucht, andernfalls die Tabellenbausteine im angegebenen Element und in allen darunter liegenden Ordnern und Netzwerken. Alternativ kann statt eines Netzwerks oder Ordners auch der Pfad einer Tabelle eingegeben werden. Dann wird der angegebene Wert nur in dieser Tabelle gesucht.
- Groß-/Kleinschreibung: bisher ist nur das zur eM-Plant-Standardmethode „finden“ analoge Verhalten implementiert: Als Übereinstimmungen gelten auch Einträge, die sich in der Groß-/Kleinschreibung vom Suchbegriff unterscheiden.
- Art der Übereinstimmung: bisher ist nur die Suche nach exakten Übereinstimmungen implementiert.
- Nur Klassen anzeigen: Wird „ja“ gewählt, so wird nur in Tabellen gesucht, die ihren Inhalt nicht erben.
- Zu durchsuchender Bereich: Bisher ist es nicht möglich, bei einer einzelnen zu durchsuchenden Tabelle den Suchbereich einzuschränken.
- Suche: Die Betätigung dieses Buttons startet die Suche nach Tabelleneinträgen, die Übereinstimmungen mit den zuvor eingegebenen Kriterien aufweisen. Das Ergebnis der Suche wird in einer Tabelle (Abbildung 23 - rechts) ausgegeben, in der die Koordinaten der jeweiligen Fundstelle (Verweis auf die Tabelle, Spalte, Zeile) sowie die dort vorgefundenen Einträge aufgeführt sind.

2.2.2.10 TabellenFilter



Dieser Baustein bietet unterschiedliche Methoden an, die nach einzelnen oder kombinierten Abfrageschlüsseln Datensätze aus eM-Plant-Tabellen herausfiltern können: eine performanceoptimierte Methode in Anlehnung an die Zugriffsmethoden des Da-

tenbausteins (*Get_Select / Get_Data_Select*) und eine flexiblere Methode, die auch komplexere Abfragen mit Vergleichsoperatoren und verschachtelten Schlüsseln zulässt (*findeDatensätze*).

Da dieser Baustein kein eigenes GUI anbietet, werden an dieser Stelle nur die öffentlichen Methoden beschrieben:

Get_Select, Get_Data_Select

Die Methoden *Get_Select* und *Get_Data_Select* haben im wesentlichen das Verhalten der gleichnamigen Datenlesezugriffsmethoden im Datenbaustein-Objekt (2.2.3.1): Mit der *Get_Select*-Methode wird eine formatierte Tabelle generiert, in die der Abfrageschlüssel einzutragen ist, um mit der *Get_Data_Select*-Methode gezielt bestimmte Datensätze aus der Datentabelle auszulesen. Die Datentabelle muss auch für diese Methoden Spaltenindizes vom Datentyp string haben.

Es gibt aber auch Unterschiede gegenüber den gleichnamigen Methoden im Datenbaustein:

1. Angabe der Datentabelle: Beim Datenbaustein wird eine Kombination aus zwei strings (Name Datenkomplex + Name Datentabelle) übergeben, beim Tabellenfilter der Verweis auf die Datentabelle (Objektpfad oder lokale/globale Variable)
2. Zeilenindex in Datentabelle: Die betrachtete Datentabelle darf Zeilenindizes haben, und diese können ebenfalls als Suchschlüssel verwendet werden.
3. Auch Werte in Spalten, die als Schlüssel definiert sind, können in der Ergebnistabelle zurückgegeben werden, sofern für die Spalte in der Schlüsseltabelle unter Indexauswahl mit „true“ gekennzeichnet wurde.

Die Kombination *Get_Select / Get_Data_Select* ist eine schnelle Methode, die sich anbietet, wenn als Suchkriterium die exakte Übereinstimmung bestimmter Datenfelder mit einem Schlüssel gilt.

FindeDatensätze

Für diese Methode kann der Abfrageschlüssel freier formuliert werden. Zum einen besteht die Möglichkeit, Kriterien mit diversen Vergleichsoperatoren zu verwenden, die in Tabelle 4 aufgeführt sind. Es können mehrere dieser Ausdrücke aneinander gereiht und mit UND bzw. ODER - Ausdrücken kombiniert werden.

Operator	Beschreibung	Datentyp
=	Eintrag in Tabelle muss mit angegebenem Wert übereinstimmen	Any
<=	Eintrag in Tabelle ist kleiner oder gleich angegebener Wert	Any
<	Eintrag in Tabelle ist kleiner angegebener Wert	Any
>	Eintrag in Tabelle ist größer angegebener Wert	Any
>=	Eintrag in Tabelle ist größer oder gleich angegebener Wert	Any
/=	Eintrag in Tabelle ungleich angegebener Wert	Any
Enthält	Eintrag in Tabelle enthält angegebenen Wert	String
Enthält nicht	Eintrag in Tabelle enthält nicht angegebenen Wert	String
Beginnt mit	Eintrag in Tabelle beginnt mit angegebenem Wert	String
Beginnt nicht mit	Eintrag in Tabelle beginnt nicht mit angegebenem Wert	String
Ist enthalten in	Eintrag in Tabelle ist in angegebenem Wert enthalten	String
ist nicht enthalten in	Eintrag in Tabelle ist nicht in angegebenem Wert enthalten	String

Endet mit	Eintrag in Tabelle endet mit angegebenem Wert	String
Endet nicht mit	Eintrag in Tabelle endet nicht mit angegebenem Wert	String

Tabelle 4 - Tabellenfilter: Operatoren für Methode *findeDatensätze*

Get_ListOfItems

Diese Methode gibt eine Liste aller in einer Tabellenspalte vorkommenden Einträge zurück (mehrfach vorkommende Einträge werden nur einmal aufgeführt). Als Parameter werden eine Tabelle (Objektpfad oder lokale/globale Variable) und eine Spalte (Spaltennummer oder Index) erwartet.

Get_TableKeys

Übergeben wird eine Tabelle und eine Liste der in dieser Tabelle zu betrachtenden Spalten. Die Methode wertet die Einträge in den angegebenen Spalten der Tabelle aus und gibt eine Tabelle zurück, die alle gefundenen Kombinationen von Werten in diesen Spalten enthält. Mehrmals vorkommende Kombinationen werden nur einmal aufgeführt.

Set_Data

Die Methode führt einen Schreibzugriff auf eine Datentabelle aus. Pro Aufruf kann ein Datensatz übergeben werden. Bestehende Datensätze passenden Schlüssels werden überschrieben, ansonsten wird der Datensatz neu angelegt.

Set_Data_Select

Die Methode führt einen Schreibzugriff auf eine Datentabelle aus. Mit einem Methodenaufruf können mehrere Datensätze übergeben werden. Die Methode Set_Data wird dann mehrmals aufgerufen. Bestehende Datensätze passenden Schlüssels werden überschrieben, ansonsten werden neue Datensätze angelegt.

Del_Data_Select

Die Methode entfernt aus einer Tabelle die zu einem angegebenen Schlüssel passenden Datensätze. Zu übergeben ist die Datentabelle (Objekt oder lokale/globale Variable), aus der Datensätze entfernt werden sollen, und eine Tabelle, die den Schlüssel enthält. Die Schlüssel-tabelle kann zuvor mit der Get_Select-Methode ermittelt und dann ausgefüllt werden. Alle Datensätze, die mit dem angegebenen Schlüssel übereinstimmen, werden entfernt.

2.2.2.11 Allgemeine Funktionen



Allgemeine Funktionen und Methoden, die keinem der übrigen Bausteine des Baustoolsets eindeutig zugeordnet werden können, sind in einem Ordner mit dem Namen „F“ zusammengetragen. Es handelt sich dabei u.a. um verschiedene Textkontrollfunktionen, die nicht zu den Standardmethoden von eM-Plant gehören, weiter um Methoden zur Konvertierung zwischen den in eM-Plant unterstützten Datentypen, Tabellenzugriffsoperationen u.a.

Es folgt eine Kurzbeschreibung der öffentlichen Methoden des Bausteins:

- **str_to**: Umwandeln eines strings in beliebigen (skalaren) Datentyp. Die Methode erzeugt keinen Fehler, wenn der string einen für den gewünschten Datentyp nicht zulässigen Wert hat. Statt dessen wird der Standardwert des jeweiligen Datentyps zurückgegeben, z.B. 01.01.1970 oder void.
- **get_table_value**: stark vereinfachte Suchfunktion für Tabellen

- `string_to_datetime`: yyyyymmddhhmmsssss-codierten string in datetime umwandeln
- `string_to_table`: Gibt die eM-Plant-Tabelle zurück, auf die der in Textform übergebene Parameter verweist
- `datetime_to_string`: datetime in yyyyymmddhhmmsssss-codierten string umwandeln
- `time_to_string`: time-Wert in string mit Angabe der Einheiten umwandeln, Anwendung: z.B. Beschriften von Gantt-Charts; z.B. wird 1:00:59:59.9000 in "1d59m59s" umgewandelt (Stundenangabe "0h" unterdrückt)
- `pos_last`: analog zu Standardmethode "pos"; Ermitteln der Position, an der <string1> LETZTMALIG in <string2> vorkommt. Falls er gar nicht vorkommt, ist das Resultat Null.
- `set_table_value`: stark vereinfachte Schreibfunktion für Tabellen
- `get_table_left`: Liefert 0, wenn eine Tabelle einen Zeilenindex hat, und 1, wenn sie keinen hat (=Nummer der ersten Spalte der Tabelle).
- `get_table_top`: Liefert 0, wenn eine Tabelle einen Spaltenindex hat, und 1, wenn sie keinen hat (=Nummer der ersten Zeile der Tabelle).
- `set_double_slash`: Fügt in einen als string angegebenen Dateipfad bei jedem Backslash („\“) einen weiteren Backslash ein. Anwendung: Umwandlung von Dateipfaden in eM-Plant-Notation
- `Datenpaket_Eintrage`: Erzeugt Eintrag in einer Datenpaket-Tabelle (2.2.9.3). Diese Methode wird im Bereich der Kommunikation zwischen den Toolsetbausteinen häufig verwendet.

2.2.3 Datenhaltung und -schnittstellen

In diesem Abschnitt werden die Bausteine vorgestellt, die der Bereitstellung simulationsrelevanter Daten dienen. Es werden die grundlegenden Tabellenstrukturen, Zugriffsmethoden und Schnittstellen beschrieben. Weiter werden die Tabellenstrukturen gezeigt, mit denen die in 2.1 beschriebenen drei Säulen der Datenmodellierung abgebildet werden.

2.2.3.1 Datenbaustein



Der Datenbaustein dient als universell verwendbarer Datenspeicher auf der Simulationsmodellseite. Er stellt die manuell eingetragenen oder aus externen Quellen (z.B. Datenbank) gewonnenen Daten den Simulationsanwendungen zur Verfügung.

Der Baustein speichert die für die schiffbauliche Fertigung notwendigen Informationen zu den Fertigungsaufträgen, Arbeitsplänen, Prozessen und Ressourcen. Seine Grundstruktur wurde so angelegt, dass eine Verwendung zum Speichern anderer Informationen wie Produktdaten, Zeit- und Schichtsysteme mit geringem Aufwand ermöglicht werden kann. So wurden bereits einige Toolsetelemente wie der Pausenbaustein (2.2.6.1) mit einem eigenen Datenbaustein versehen.

2.2.3.1.1 Aufbau und Funktionen

Dieser Abschnitt enthält die Spezifikationen für den Datenbaustein des „Simulationstoolset Schiffbauliche Fertigung“ der Meyer Werft. Es beschreibt den Aufbau und die Funktion der relevanten Elemente des Hauptordners und der Unterordner.

Fachklasse „Datenbaustein“

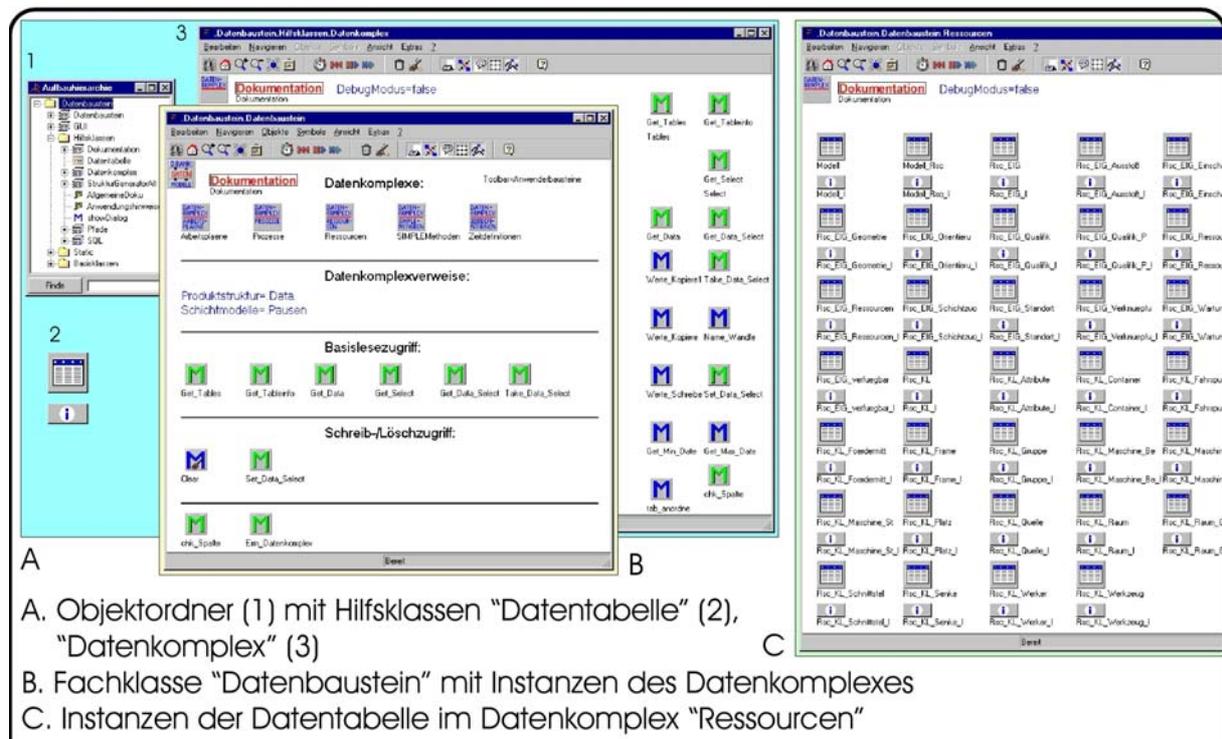
Das Netzwerk „Datenbaustein“ (Abbildung 24-B) ist die Fachklasse und zugleich das zentrale Element zum Speichern von Produktions- und Ressourcendaten. In der Fachklasse werden Instanzen der Hilfsklasse „Datenkomplex“ eingesetzt, die - thematisch gegliedert - die relevanten Datentabellen aufnehmen.

Der Datenbaustein enthält weiterhin öffentliche Zugriffsmethoden. Sie kommunizieren mit gleichnamigen Methoden in den Datenkomplexen und dienen dem Lese- und Schreibzugriff auf die dortigen Tabellen (siehe Beschreibung der Methoden in Abschnitt 2.2.3.1.3).

Hilfsklasse „Datentabelle“

Die Hilfsklasse „Datentabelle“ bildet die Grundlage für alle Daten- und Informationstabellen der Datenkomplexe. Alle zu speichernden Daten werden in den Datentabellen hinterlegt. Jeder Datentabelle ist eine Informationstabelle mit der Endung „_I“ zugeordnet, in der alle Informationen in Bezug auf Anzahl der Spalten, Spaltenindizes, Datentyp, Schlüsselfunktion und Beschreibung der Spalte der Datentabelle hinterlegt sind. Weiterhin sind die ausgeschriebene Tabellenbezeichnung und eine Kurzbeschreibung der Tabelle im Tabellenkommentar angegeben.

Hilfsklasse „Datenkomplex“



A. Objektordner (1) mit Hilfsklassen "Datentabelle" (2), "Datenkomplex" (3)

B. Fachklasse "Datenbaustein" mit Instanzen des Datenkomplexes

C. Instanzen der Datentabelle im Datenkomplex "Ressourcen"

Datenbaustein: Struktur



Abbildung 24 - Datenbaustein: Struktur

Der Datenkomplex speichert die Informationen eines Datengebiets in verschiedenen Datentabellen. In jedem Datenbaustein können ein oder mehrere Datenkomplexe instanziiert werden. Abbildung 24-C zeigt den Datenkomplex „Ressourcen“ des Datenbausteins und die instanziierten Daten- und Informationstabellen..

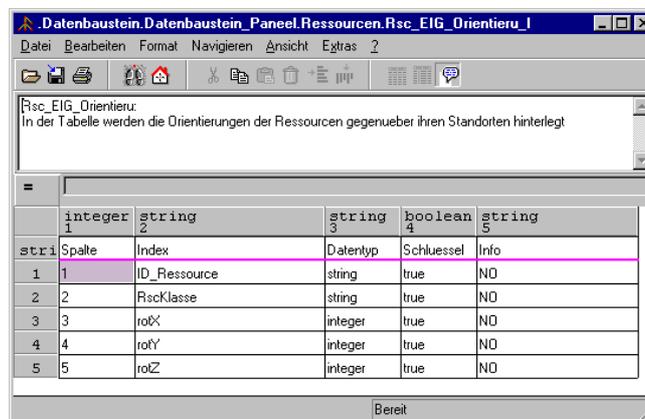
Auf die Datenkomplexe kann über die öffentlichen Methoden zugegriffen werden, die namentlich mit denen des Datenbausteins übereinstimmen. Die Funktion der öffentlichen Methoden wird in Abschnitt näher erläutert.



	string 1	string 2	integer 3	integer 4	integer 5
stri	ID_Ressource	RscKlasse	rotX	rotY	rotZ
1	Platz1	Platz	0	0	90
2	Platz2	Platz	0	0	0
3	Quelle	Quelle	0	0	0
4					
5					
6					

Abbildung 25 - Datentabelle: Instanz Datentabelle Rsc_EIG_Orientieru

Abbildung 25 zeigt die Datentabelle „Rsc_EIG_Orientieru“, ihre Informationstabelle ist in Abbildung 26 dargestellt.



Rsc_EIG_Orientieru:
In der Tabelle werden die Orientierungen der Ressourcen gegenüber ihren Standorten hinterlegt

	integer 1	string 2	string 3	boolean 4	string 5
stri	Spalte	Index	Datentyp	Schlüssel	Info
1	1	ID_Ressource	string	true	NO
2	2	RscKlasse	string	true	NO
3	3	rotX	integer	true	NO
4	4	rotY	integer	true	NO
5	5	rotZ	integer	true	NO

Abbildung 26 - Datentabelle: Instanz Informationstabelle Rsc_EIG_Orientieru

GUI

Über das Netzwerk „GUI“ können per grafischer Bedienoberfläche Daten aus einer Datenbank in die Tabellen des Bausteins transferiert oder Simulationsergebnisse etc. in die Datenbank gespeichert werden (Abbildung 27).

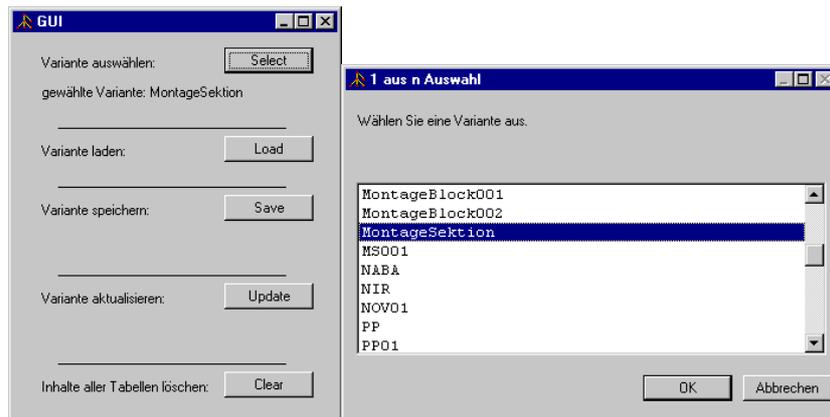


Abbildung 27 - Datenbaustein: GUI

2.2.3.1.2 Datenstruktur

Dieser Dokumentabschnitt enthält die Spezifikationen der Datenstruktur für das „Simulationsstoolset Schiffbauliche Fertigung“ der Meyer Werft.

Die Datenstruktur wurde unter Verwendung einer Datenbank entwickelt und bildet alle als relevant erkannten Datenkomplexe (Als Datenkomplex werden alle einem Datenbereich zugeordneten Tabellen bezeichnet) ab.

Im folgenden werden die einzelnen Datenkomplexe näher erläutert und spezifiziert.

Arbeitspläne

Der Arbeitsplan enthält alle Angaben über die zur Produkterstellung notwendigen Arbeitsschritte. Seine Struktur ist in Abbildung 28 wiedergegeben. Im wesentlichen geben die hier zu modellierenden Daten Auskunft über die Fragen:

- Welche Arbeitsschritte sind definiert?
- Wie lauten die Anordnungsbeziehungen der Arbeitsschritte untereinander?
- Welche Bedingungen sind bei verknüpften Arbeitsschritten zu beachten?
- Welche Parameter beeinflussen die Dauer der einzelnen Arbeitsschritte?
- Welchen Arbeitsschritten wurden bereits Ressourcen zugeordnet?
- Welche Arbeitsschritte sind Montageschritte, bei denen Bauteile übergeordneten Baugruppen zugeführt werden?
- Wie sind die Arbeitsschritte zueinander hierarchisch angeordnet?

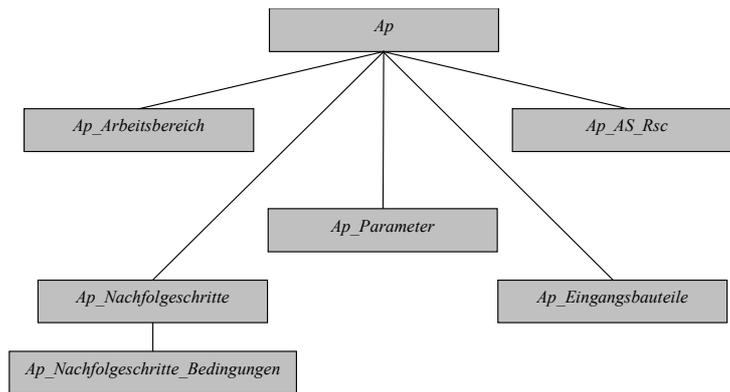


Abbildung 28 - Datenbaustein: Arbeitsplanstruktur

Prozesse

Prozesse sind neben den Qualifikationen das Bindeglied zwischen den im Arbeitsplan festgelegten Arbeitsschritten und den Arbeitsressourcen. Jedem Arbeitsschritt ist in der Ap-Tabelle ein Prozess zugeordnet. Zur Ausführung eines Prozesses sind eine oder mehrere Qualifikationen erforderlich (Tabelle Proc_Qualifik), die wiederum von Arbeitsressourcen angeboten werden. Die Datenstruktur der Prozesse ist in Abbildung 29 wiedergegeben.

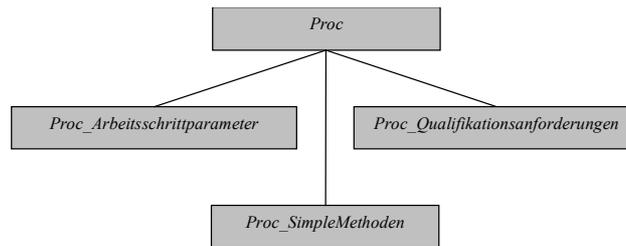


Abbildung 29 - Datenbaustein: Prozessstruktur

Prozesszeitermittlung

Die Berechnung der Prozesszeit kann durch eine in der Datenstruktur zu hinterlegende Prozessauswertemethode durchgeführt werden. Diese bezieht ihre Eingangsparameter für den aktuellen Arbeitsschritt aus dem Arbeitsplan und den globalen Ressourceneigenschaften (siehe Abbildung 30).

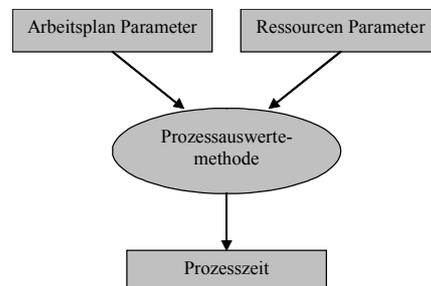


Abbildung 30 - Datenbaustein: Prozesszeitermittlung

Ressourcen

Über die Ressourcen (Abbildung 31) wird die Simulationsumgebung abgebildet. Sie stellen die für Prozesse benötigten Qualifikationen zur Verfügung und/oder nehmen Bauteile auf.

Klassen	Eigenschaften
<i>Rsc_KL_Foerdermittel</i>	<i>Rsc_EIG_Einschaltdauer</i>
<i>Rsc_KL_Frames</i>	<i>Rsc_EIG_Geometrie</i>
<i>Rsc_KL_Maschine_Beweglich</i>	<i>Rsc_EIG_Orientierung</i>
<i>Rsc_KL_Maschine_Fest</i>	<i>Rsc_EIG_Qualifikationen</i>
<i>Rsc_KL_Plaetze</i>	<i>Rsc_EIG_Qualifikationen_Parameter</i>
<i>Rsc_KL_Quellen</i>	<i>Rsc_EIG_Ressource_halten</i>
<i>Rsc_KL_Raum</i>	<i>Rsc_EIG_Ressourcenzuordnung</i>
<i>Rsc_KL_Raum_gesperrt</i>	<i>Rsc_EIG_Schichtzuordnung</i>
<i>Rsc_KL_Schnittstellen</i>	<i>Rsc_EIG_Standort</i>
<i>Rsc_KL_Senken</i>	<i>Rsc_EIG_Verknuepfungen</i>
<i>Rsc_KL_Werker</i>	<i>Rsc_EIG_Wartung</i>
<i>Rsc_KL_Werkzeug</i>	

Abbildung 31 - Datenbaustein: Ressourcenstruktur

Es existieren zwei Typen von Ressourcentabellen. Während die Eigenschaftstabellen jeweils einen Eigenschaftstypus für verschiedene Ressourcen speichern können, werden in den Klassentabellen die Ressourcen selbst und ihre typischen Eigenschaften hinterlegt. Eigenschaftstabellen können, aber müssen nicht mit Daten gefüllt werden.

Die Nomenklatur der Ressourcenbezeichnung entspricht dem folgenden Beispiel:

- Klassentabelle: `Rsc_KL_Werker`
- Eigenschaftstabelle: `Rsc_EIG_Geometrie`

Die Ressourcen unterschiedlicher Klassen müssen nicht über eindeutige Namen verfügen, da über einen Klassenschlüssel eine Unterscheidung ermöglicht wird.

SimpleMethoden

Der Datenkomplex „SimpleMethoden“ enthält alle für die Prozesszeiterrechnung oder Unterbrechungsermittlung notwendigen Methoden in Form von Textstrings.

2.2.3.1.3 Öffentliche Methoden

Es existieren verschiedene Schnittstellenmethoden, denen ein oder mehrere, die jeweilige Abfrage spezifizierende Parameter übergeben werden und die jeweils eine Tabelle mit dem Abfrageergebnis zurückliefern. Grundsätzlich ist jede der im folgenden beschriebenen Zugriffsmethoden in der Hilfsklasse „Datenkomplex“ implementiert. Es existiert aber jeweils eine gleichnamige Methode im Datenbaustein, die unter Angabe eines Datenkomplexes aufgerufen werden kann. Die Anfrage wird an die Methode in der entsprechenden Instanz des Datenkomplexes weitergeleitet. Im folgenden werden nur die Zugriffsmethoden in der Klasse Datenkomplex erläutert.

Zugriffsmethoden

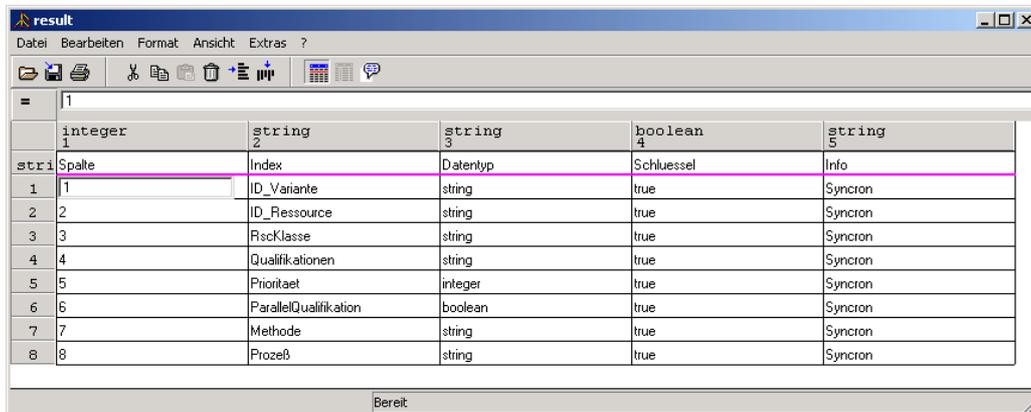
Um einen unkomplizierten und schnellen Zugriff zu ermöglichen, wurden möglichst wenige, einfache und universelle Methoden definiert, die alle relevanten Daten strukturiert zur Verfügung stellen. Sie sind für alle Datenkomplexe gleich definiert. Zunächst werden die Methoden für den Lesezugriff beschrieben. Die Schreibzugriffsmethoden basieren auf diesen Methoden.

Get_Tables

Get_Tables gibt die ausgeschriebenen Tabellenbezeichnungen sowie eine kurze verbale Beschreibung in Form einer Tabelle zurück.

Get_Tableinfo

Die Methode gibt die in der Infotabelle hinterlegten Angaben in unveränderter Form zurück. Für einen Get_Tableinfo-Zugriff auf den Datenkomplex „Ressourcen“, Tabelle „Rsc_EIG_Qualifik“ ergibt sich z.B. die in Abbildung 32 dargestellte Tabelle.



	integer 1	string 2	string 3	boolean 4	string 5
str1	Spalte	Index	Datentyp	Schluessel	Info
1	1	ID_Variante	string	true	Syncron
2	2	ID_Ressource	string	true	Syncron
3	3	RscKlasse	string	true	Syncron
4	4	Qualifik-ationen	string	true	Syncron
5	5	Prioritaet	integer	true	Syncron
6	6	ParallelQualifikation	boolean	true	Syncron
7	7	Methode	string	true	Syncron
8	8	Prozeß	string	true	Syncron

Abbildung 32 - Datenbaustein: Beispiel Get_Tableinfo, Rückgabetable für Tabelle Rsc_EIG_Qualifik

Get_Select

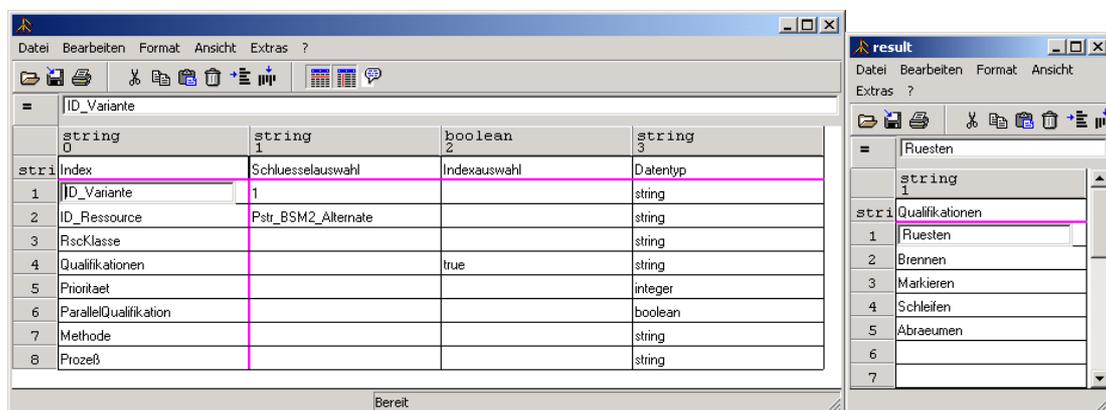
Get_Select wird im Zusammenhang mit Get_Data_Select verwendet, um die nach Schlüsseln und Indizes gefilterten Daten einer Tabelle auszulesen.

Zunächst kann über Get_Select eine Auswahltabelle angefordert werden, in der die Spaltenindizes der betrachteten Tabelle in den Zeilen der nullten Spalte aufgeführt sind. Nun kann definiert werden, welche Spalten als Zugriffsschlüssel dienen und aus welchen Werte ausgelesen werden sollen.

Um einen Index als Zugriffsschlüssel zu definieren, wird in der ersten Spalte der jeweiligen Zeile der Auswahltabelle ein Schlüssel als String hinterlegt. Die zurückzugebenden Spalten werden durch Setzen des Flags in Spalte 2 der Auswahltabelle markiert.

Die so ausgefüllte Tabelle wird an die Methode Get_Data_Select übergeben, welche wiederum die spezifizierten Daten in Tabellenform zurückliefert.

Im Beispiel in Abbildung 33 (links) ist eine ausgefüllte Auswahltabelle für einen Lesezugriff auf die Tabelle „Rsc_EIG_Qualifik“ des Datenkomplexes „Ressourcen“ dargestellt.



	string 0	string 1	boolean 2	string 3
str1	Index	Schluesselauswahl	Indexauswahl	Datentyp
1	ID_Variante	1		string
2	ID_Ressource	Pstr_BSM2_Alternate		string
3	RscKlasse			string
4	Qualifikationen		true	string
5	Prioritaet			integer
6	ParallelQualifikation			boolean
7	Methode			string
8	Prozeß			string

	string 1
str1	Qualifikationen
1	Ruesten
2	Brennen
3	Markieren
4	Schleifen
5	Abraeumen
6	
7	

Abbildung 33 - Datenbaustein: Beispiel Get_Select / Get_Data_Select, Auswahl- und Rückgabetable für Zugriff auf Tabelle Rsc_EIG_Qualifik

Get_Data_Select

Die Methode `Get_Data_Select` benötigt die über `Get_Select` angeforderte und entsprechend ausgefüllte Auswahltabelle sowie den Namen der betrachteten Datentabelle als Parameter und liefert die spezifizierten Daten in Form einer Tabelle zurück.

Die Übergabe der ausgefüllten `Get_Select`-Tabelle an die Methode `Get_Data_Select` führt zur Rückgabe der in Abbildung 33 (rechts) gezeigten Tabelle.

Set_Data_Select

Für den Schreibzugriff auf die Tabellen des Datenbausteins wurde ein dem `Get_Select/Get_Data_Select` - Mechanismus ähnliches Verfahren gewählt. Mit `Get_Select` wird eine Auswahltabelle generiert, in die dann der Schlüssel für den Datensatz eingetragen wird, der angelegt oder überschrieben werden soll. Die zu schreibenden Daten werden in eine zusätzlich zu übergebende Wertetabelle eingetragen, die an die Methode `Set_Data_Select` übergeben wird. Optional kann die Wertetabelle mehrere Zeilen enthalten, so dass mit einem Aufruf mehrere Datensätze verändert werden. In diesem Fall muss ein weiterer Parameter angeben, welches Datenfeld in der Wertetabelle als Erweiterung des Zugriffsschlüssels zu betrachten ist.

Take_Data_Select

Diese Methode bietet bei Tabellen mit vielen (>1000) Datensätzen einen schnelleren Lesezugriff als `Get_Data_Select`. Die höhere Performance beruht auf einer Vorsortierung der Datentabelle nach einer ihrer Spalten, dem sogenannten Suchindex.

Ist der Suchindex Bestandteil des Schlüssels eines Lesezugriffs mit der `Take_Data_Select`-Methode, so kann der Suchaufwand erheblich reduziert werden, indem von vornherein nur der Bereich der Datentabelle mit dem Schlüssel verglichen wird, in dem die Werte des Suchindex mit dem Abfrageschlüssel verglichen werden.

2.2.3.2 Data-Baustein



Im Data-Baustein werden in der Simulationsumgebung die die Produktstruktur betreffenden Informationen einer Planungsvariante gespeichert und mit speziellen öffentlichen Methoden der Simulation zugänglich gemacht. Mittels einer SQL-Schnittstelle können Daten aus einer Datenbank eingelesen sowie dort gespeichert werden.

2.2.3.2.1 Aufbau und Funktion

Abbildung 34 zeigt die wesentlichen Klassen des Data-Bausteins.

- Fachklasse: Die Fachklasse (Netzwerk „Data“) enthält die Datentabellen, auf die von seiten der Simulationsumgebung zugegriffen wird, und die dafür verwendeten öffentlichen Methoden (Beschreibung s.u.).
- Die Schnittstelle zur Datenbank ist mit den beiden Klassen „SQL“ und einem Datenbaustein (s.a. 2.2.3.1) realisiert.
- Die Benutzeroberfläche ist in der Klasse „GUI“ modelliert.

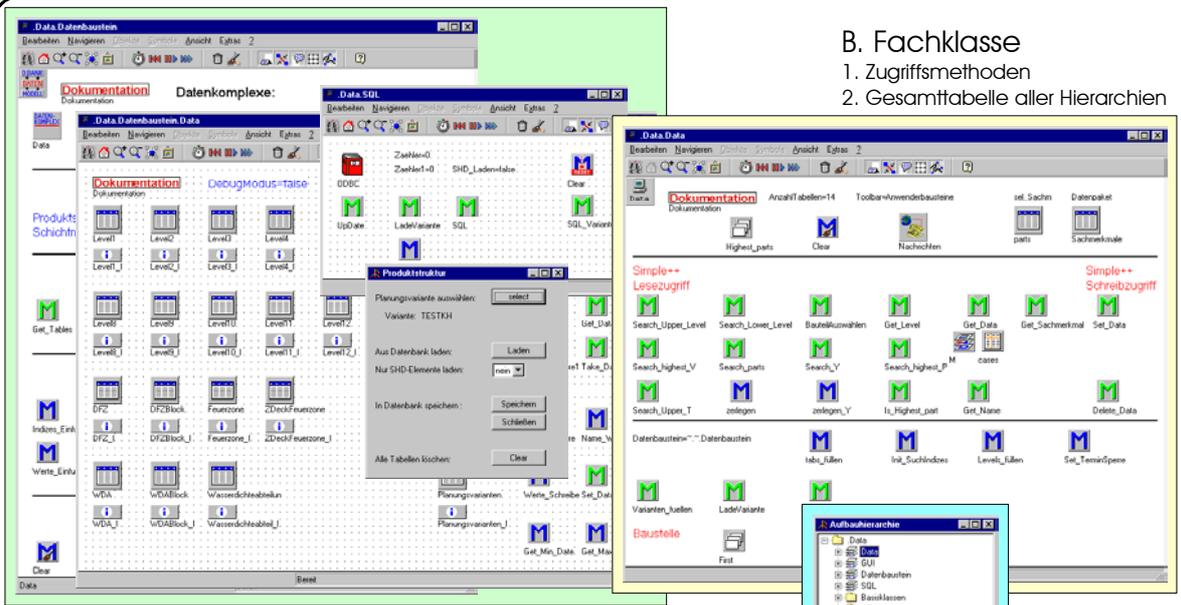
2.2.3.2.2 Datenstruktur

- Fachklasse: Da der Data-Baustein in einer frühen Projektphase entwickelt wurde, in der noch nicht das allgemeine Datenmodell und der Datenbaustein zur Verfügung standen, wurde zunächst die Gesamttabelle „parts“ in der Fachklasse angelegt. In sie werden die

G:\Abschluss\Bericht\Gesamtbericht v03.11.doc

Elemente aus allen Hierarchien der Produktstruktur eingetragen, sofern sie sich auf die ursprünglich ausschließlich betrachtete Sicht „Baufolge“ (Hierarchy of Assembly) beziehen. In einer zusätzlichen Sachmerkmaltabelle werden zusätzliche Eigenschaften der Elemente eingetragen.

- Datenbaustein: Abweichend von der oben beschriebenen Datenstruktur werden in der Datenbank Einzeltabellen für jede Ebene der Hierarchie geführt. Außerdem sind weitere Tabellen implementiert, die zusätzliche Topologiesichten abdecken. Um allgemeine SQL-Funktionen für den Datentransfer verwenden zu können und Daten zu allen Sichten für die Simulation bereitzustellen, wurde der Datenbaustein in den Data-Baustein integriert und dort die Tabellenstruktur der Datenbank nachgebildet.



B. Fachklasse

1. Zugriffsmethoden
2. Gesamttabelle aller Hierarchien

C. Weitere Klassen

1. Datenbaustein mit eigenen Datentabellen je Hierarchieebene und SQL-Netzwerk als Schnittstellen zur Datenbank
2. GUI

A. Struktur

Data-Baustein: Aufbau



Abbildung 34 - Data-Baustein: Aufbau

2.2.3.2.3 Öffentliche Methoden

Für den Lesezugriff auf die in den Datentabellen gespeicherten Informationen werden folgende öffentliche Methoden zur Verfügung gestellt:

Abfragen von Informationen zu einem bestimmten Bauteil

Informationen zu einem gegebenen Bauteil werden mit folgenden Methoden abgefragt. Wenn nicht anders angegeben, wird als Parameter der Schlüssel des Bauteils (ID_Topologie) erwartet.

- **Get_Data**

Allgemeine Methode zum Abfragen von Bauteilinformationen. Neben dem Bauteilschlüssel muss noch ein Schlüsselwort „wkod“ (what kind of data) übergeben werden, Rückgabewert ist eine Liste. Jedem gültigen wkod ist eine Methode zugeordnet, die be-

stimmt Informationen aus der Tabelle „parts“ ausliest und in die Liste einträgt. Zum Beispiel werden die Werte der Spalten „Länge“, „Breite“ und „Höhe“ eingetragen, wenn das Schlüsselwort „Abmessungen“ übergeben wurde.

- **Get_Level**

Ermittelt zu einem angegebenen Element der Produktstruktur dessen Position in der Hierarchy of Assembly. Zurückgegeben wird ein String, z.B. „Level1“ für einen Kostenträger.

- **Get_Sachmerkmal**

Liest Sachmerkmal(e) eines Bauteils aus der Sachmerkmale-Tabelle aus. Zu übergeben ist neben dem Bauteilschlüssel der Bezeichner eines Sachmerkmals, nach dem gezielt gesucht werden soll. Sollen alle Sachmerkmale eines Bauteils zurückgegeben werden, so ist ein Leerstring zu übergeben.

- **Get_Name**

Gibt den „Namen“ eines Bauteils zurück. Unter dem Namen wird hier das letzte Element des Schlüssels des Bausteins (ID_Topologie) verstanden, z.B. lautet bei einem Bauteil mit der ID_Topologie „MontageSektion_649_1_07_2421B“ der Name „2421B“. Die Methode liest übrigens nicht aus einer Datentabelle aus, sondern führt lediglich eine Textkontrollfunktion mit dem übergebenen Bauteilschlüssel aus.

- **Is_Highest_part**

Prüft, ob das angegebene Bauteil in der Hierarchy of Assembly an oberster Stelle (Level1) steht.

Ermitteln von Bauteilen

Die folgenden Methoden geben jeweils ein Bauteil oder eine Liste von Bauteilen zurück, die bestimmte Kriterien erfüllen.

- **Search_Upper_Level**

Ermittelt zu einem gegebenen Bauteil das in der Hierarchy of Assembly übergeordnete Element.

- **Search_Lower_Level**

Ermittelt zu einem gegebenen Bauteil die in der Hierarchy of Assembly untergeordneten Bauteile. Zurückgegeben wird eine Liste mit n Einträgen.

- **Search_Highest_V**

Ermittelt zu einem gegebenen Bauteil, das in der Datentabelle als fremdvergeben markiert ist (Datenfeld „Vergabe“) das höchste übergeordnete Bauteil, das ebenfalls vergeben ist.

- **Search_parts**

Ermittelt zu einem gegebenen Bauteil alle Einzelteile, die keine weiteren untergeordneten Teile haben. Rückgabewert ist eine Liste mit diesen Teilen.

- **Search_Y**

Ermittelt zu einem Bauteil, bei dem in der Datentabelle im Datenfeld „Key_Schedule“ der Wert „X“ eingetragen ist, alle in den untergeordneten Levels enthaltenen Bauteile mit dem Eintrag „Y“.

- **Search_highest_P**

Ermittelt alle Bauteile der geladenen Planungsvariante, zu denen kein übergeordnetes Bauteil in der Produktstruktur eingetragen ist.

2.2.3.3 SQL-Baustein



Der SQL-Baustein bildet eine allgemeine Schnittstelle, mit der Daten zwischen einer Datenbank und den in den Simulationsbausteinen verwendeten Datentabellen (Datenbausteinen) transferiert werden können. Die Datentabellen können gelesen, geschrieben und aktualisiert werden.

2.2.3.3.1 Anforderungen

Jeder Relation in der Datenbank kann in der Simulationsumgebung eine Tabelle zur Aufnahme der Daten entsprechen. Eine Aufteilung einer Datenbank-Tabelle in unterschiedliche eM-Plant-Tabellen ist im allgemeinen nicht vorzusehen. Zu beachten ist, dass bei den eM-Plant Tabellen eine Strukturierung, wie sie in der Datenbank vorliegt, nicht vorausgesetzt werden kann. Dies bezieht sich sowohl auf die Anordnung der Feldnamen als auch auf die Primärschlüssel in der Datenbank. Die Definition der Relationen in der Datenbank ist in jedem Fall ausschlaggebend.

Beim Einlesen der Daten ist die Struktur der Datentabellen in den Modellbausteinen an die Struktur in der Datenbank anzupassen. Anschließend sind die gewünschten Daten in die Tabelle zu übertragen. Werden Daten aus der Simulationsumgebung zurückgeschrieben bzw. aktualisiert, ist auf Übereinstimmung der Strukturen zu achten. Liegt keine Übereinstimmung vor, ist die Aktion abzubrechen.

Die Auswahl der zu übertragenden Tabellen richtet sich nach der Simulationsumgebung.

2.2.3.3.2 Realisierung

Der erste Abschnitt dieses Kapitels verdeutlicht den erforderlichen Aufbau der Datentabellen und Informationstabellen, der zweite bezieht sich auf die Implementierung der Methoden, die den Datentransfer ausführen.

Datenbeschreibung

Damit die SQL Schnittstelle für unterschiedliche Anwendungen eingesetzt werden kann, müssen einige Rahmenbedingungen berücksichtigt werden. Die Bedingungen beziehen sich sowohl auf die Datenbank als auch auf den Aufbau der Tabellen in eM-Plant.

- **Access Datenbank**

Der Zugriff auf die Datenbank erfolgt über das von eM-Plant gelieferte ODBC-Objekt. Daher ist die Einrichtung einer ODBC-Benutzerdatenquelle erforderlich. Nach der Initialisierung der Tabellen im Simulationsmodell darf diese Quelle nicht verändert werden. Es ist nicht erforderlich, dass die Datenquelle auf jedem Simulationsrechner gleich benannt wird.

Die Datenbank muss eine Tabelle „**Datenstruktur**“ mit dem in Abbildung 35 dargestellten Aufbau beinhalten. Zur Pflege dieser Tabelle ist nach jeder Änderung des in der Datenbank enthaltenen Schema das Makro „**Strukturerzeugen**“ auszuführen. Das Modul „**Strukturre-**

port“ aktualisiert daraufhin den Inhalt der Tabelle. Die Tabelle „Datenstruktur“ ist die Grundlage für jede Information, die im Modell über die Datenbank vorliegt.

ID	Tabelle	TabelleBeschreibung	Index	Datentyp Datenbank	Datentyp SIMPLE++	Beschreibung	PKey

Abbildung 35 - SQL-Baustein: Schema der Tabelle „Datenstruktur“

- **eM-Plant**

In den eM-Plant Objekten werden die eingelesenen Daten in Datentabellen gesichert. Die Datentabellen werden mit den Namen der entsprechenden Access-Tabelle bezeichnet, maximal ist der Name jedoch achtzehn Zeichen lang. Die Informationstabellen bestimmen die in den zugeordneten Datentabellen enthaltenen Daten. Sie sind einer Datentabelle über den Namen der Datentabelle mit angehängten „_I“ zugeordnet und müssen sich mit den Datentabellen im selben Netzwerk befinden.

Maßgebend für die Struktur der beiden Tabellen ist die Datentabelle „Datenstruktur“ in der AccessDatenbank.

Dem Kommentar der Informationstabelle sind drei Angaben zu entnehmen, die jeweils in einer Zeile abgelegt werden:

1. Zeile: Name der AccessTabelle, deren Inhalt in die Datentabelle gelesen werden soll
2. Zeile: Beschreibung der Tabelle
3. Zeile: Bezeichner der ODBC-Benutzerdatenquelle

Der Kommentar wird in der Methode „ErzeugeStruktur“ erzeugt und bei keiner anderen Operation verändert. Auch beim Erzeugen einer neuen Datenstruktur, z. B. beim Einlesen einer neu strukturierten Tabelle wird die Information nicht verändert!

Der Inhalt einer Informationstabelle ist in Abbildung 36 angegeben. Von besonderer Bedeutung sind die Spalten fünf bis sieben. In der Spalte „Info“ wird durch das Schlüsselwort „Synchron“ angegeben, dass der Inhalt des Feldes bei einer Aktualisierung mit verändert wird. Ändert der Anwender Werte in dieser Spalte, wird das Feld bei der Aktualisierung der Datensätze nicht berücksichtigt.

Die Spalte „Pkey“ zeigt an, ob ein Feld in der Access-Datenbank zum Primärschlüssel gehört oder nicht.

Sollen nicht alle Daten aus der Datenbank geladen werden, die zu einer Tabelle gehören, kann in der Spalte „Wertebereich“ eine Einschränkung getroffen werden. Das in Abbildung 5 gegebene Beispiel schränkt die einzulesenden Datensätze auf Sätze ein, die im Datenfeld „ID_Variante“ die Bezeichnung „MiniModell“ haben. Werden in mehreren Feldern Angaben zur Einschränkung getroffen, sind diese durch ein logisches „UND“ verknüpft. Sollen „ODER“-Verknüpfungen getroffen werden, sind diese explizit in der Einschränkung anzugeben.

integer	string	string	boolean	string	string	boolean
1	2	3	4	5	6	7
Spalte	Index	Datentyp	Schlüssel	Info	Wertebereich	PKey
1	ID_Variante	string	true	Synchron	ID_Variante=MiniModell	true
2	ArbeitsschrittVorgaenger	string	true	Synchron		true
3	ID_Topologie_V	string	true	Synchron		true
4	Sicht_V	string	true	Synchron		true
5	ArbeitsschrittNachfolger	string	true	Synchron		true
6	ID_Topologie_N	string	true	Synchron		true
7	Sicht_N	string	true	Synchron		true
8	ID_Bedingung_N	string	true	Synchron		false
9	Prioritaet	integer	true	Synchron		false
10	Verknuepfung	string	true	Synchron		false
11	Lag	integer	true	Synchron	Lag=10	false
12	LagProzent	real	true	Synchron		false
13	Nummer	integer	true	Synchron		false
14	ID_Eltern	string	true	Synchron		false
15	ID_Topologie_Eltern	string	true	Synchron		false
16	Sicht_Eltern	string	true	Synchron		false

Abbildung 36 - SQL-Baustein: Aufbau einer Informationstabelle

In der Abbildung 37 ist ein Beispiel für eine Datentabelle angegeben. Die grau hinterlegten Spalten sind schreibgeschützt. Die entsprechenden Felder sind Schlüsselfelder in der Datenbank. Die Spalten werden besonders behandelt, damit nicht unbemerkt neue Datensätze in die Datenbank aufgenommen werden. Würde ein Anwender einen Datensatz überschreiben, in dem ein Schlüssel verändert wurde, wird der Satz in die Datenbank neu eingefügt. Der in der Tabelle überschriebene Satz hat in der Datenbank aber weiterhin Bestand.

string	string	string	string	string	string	string	string	string	integer	string
1	2	3	4	5	6	7	8	9	10	11
ID_Variante	ID_Arbeitschritt	ID_Topologie	Sicht	Arbeitsschritt	ID_Eltern	ID_Topologie	Sicht_Eltern	Hierarch	Prozess	
DummyTest	DummyTest_Konstruktion	DummyTest_1111	DFZ	Konstruktion	DummyTe	DummyTest_1	DFZ	1	Konstruktion	1
DummyTest	DummyTest_Konstruktion_FG3	DummyTest_1111_FZ1_0	DFZ	FG3	DummyTe	DummyTest_1	DFZ	2	FG3	1
DummyTest	DummyTest_Konstruktion_FG3	DummyTest_1111_FZ1_0	DFZ	Abnahme	DummyTe	DummyTest_1	DFZ	3	Konstruktion	1
DummyTest	DummyTest_Konstruktion_FG3	DummyTest_1111_FZ1_0	DFZ	Abnahme	DummyTe	DummyTest_1	DFZ	3	Konstruktion	7
DummyTest	DummyTest_Konstruktion_FG3	DummyTest_1111_FZ1_0	DFZ	Konstruktion	DummyTe	DummyTest_1	DFZ	3	Konstruktion	7
DummyTest	DummyTest_Konstruktion_FG3	DummyTest_1111_FZ1_0	DFZ	Konstruktion	DummyTe	DummyTest_1	DFZ	3	Konstruktion	1
DummyTest	DummyTest_Konstruktion_FG4	DummyTest_1111_FZ1_0	DFZ	FG4	DummyTe	DummyTest_1	DFZ	2	FG4	1
DummyTest	DummyTest_Konstruktion_FG4	DummyTest_1111_FZ1_0	DFZ	Abnahme	DummyTe	DummyTest_1	DFZ	3	Abnahme	1
DummyTest	DummyTest_Konstruktion_FG4	DummyTest_1111_FZ1_0	DFZ	Konstruktion	DummyTe	DummyTest_1	DFZ	3	Konstruktion	1
DummyTest	DummyTest_Konstruktion_FG4	DummyTest_1111_FZ1_0	DFZ	Veigabe	DummyTe	DummyTest_1	DFZ	3	Veigabe	1
DummyTest	DummyTest_Konstruktion_FG5	DummyTest_1111_FZ1_0	ASS	FG5	DummyTe	DummyTest_1	ASS	2	FG5	1
DummyTest	DummyTest_Konstruktion_FG5	DummyTest_1111_FZ1_0	ASS	AV	DummyTe	DummyTest_1	ASS	3	AV	1
DummyTest	DummyTest_Konstruktion_FG5	DummyTest_1111_FZ1_0	ASS	AV	DummyTe	DummyTest_1	ASS	3	AV	1
DummyTest	DummyTest_Konstruktion_FG3	DummyTest_1111_FZ1_0	DFZ	FG3	DummyTe	DummyTest_1	DFZ	2	FG3	1
TYPKTR002	TYPKTR002_Aussenhaufbau	TYPKTR002_002_104	Bauplan	Aussenhaufbau(gebogen)	TYPKTR0	TYPKTR002	Bauplan	1	Aussenhaufbau	1
TYPKTR002	TYPKTR002_Aussenhaufbau	TYPKTR002_003_104	Bauplan	Aussenhaufbau(gebogen)	TYPKTR0	TYPKTR002	Bauplan	1	Aussenhaufbau	1
TYPKTR002	TYPKTR002_B_auschnittbau	TYPKTR002_001_301	Bauplan	Bauschnittbau	TYPKTR0	TYPKTR002	Bauplan	1	Bauschnittbau	1
TYPKTR002	TYPKTR002_B_auschnittbau	TYPKTR002_002_301	Bauplan	Bauschnittbau	TYPKTR0	TYPKTR002	Bauplan	1	Bauschnittbau	1
TYPKTR002	TYPKTR002_B_auschnittbau	TYPKTR002_003_301	Bauplan	Bauschnittbau	TYPKTR0	TYPKTR002	Bauplan	1	Bauschnittbau	1
TYPKTR002	TYPKTR002_B_blockbau	TYPKTR002_001_001	Bauplan	Blockbau	TYPKTR0	TYPKTR002	Bauplan	1	Blockbau	1
TYPKTR002	TYPKTR002_B_blockbau	TYPKTR002_001_002	Bauplan	Blockbau	TYPKTR0	TYPKTR002	Bauplan	1	Blockbau	1
TYPKTR002	TYPKTR002_B_blockbau	TYPKTR002_001_003	Bauplan	Blockbau	TYPKTR0	TYPKTR002	Bauplan	1	Blockbau	1
TYPKTR002	TYPKTR002_B_blockbau	TYPKTR002_001_004	Bauplan	Blockbau	TYPKTR0	TYPKTR002	Bauplan	1	Blockbau	1
TYPKTR002	TYPKTR002_B_blockbau	TYPKTR002_001_005	Bauplan	Blockbau	TYPKTR0	TYPKTR002	Bauplan	1	Blockbau	1
TYPKTR002	TYPKTR002_B_blockbau	TYPKTR002_001_006	Bauplan	Blockbau	TYPKTR0	TYPKTR002	Bauplan	1	Blockbau	1
TYPKTR002	TYPKTR002_B_blockbau	TYPKTR002_001_007	Bauplan	Blockbau	TYPKTR0	TYPKTR002	Bauplan	1	Blockbau	1

Abbildung 37 - SQL-Baustein: Inhalt einer Datentabelle mit grau hinterlegten Schlüsselfeldern

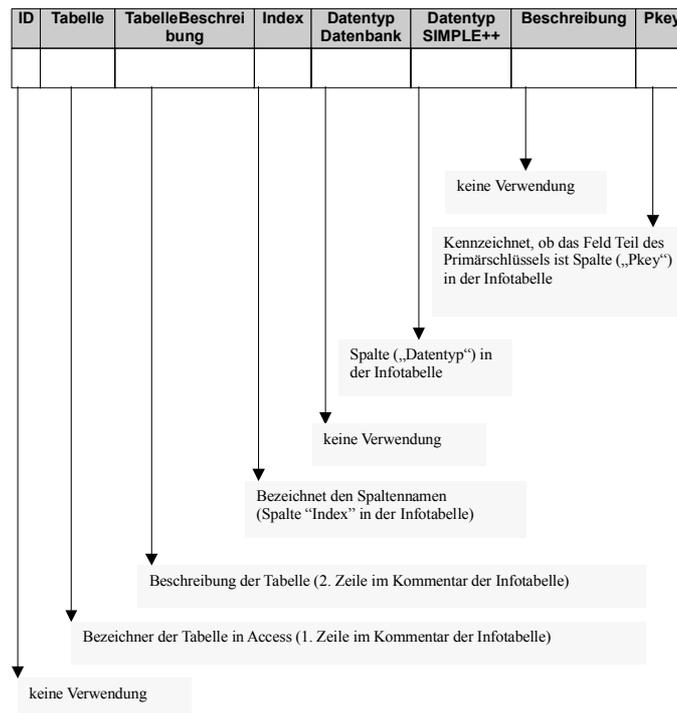


Abbildung 38 - SQL-Baustein: Zusammenhang Tabellenbeschreibung Access / Informationstabelle eM-Plant

Die Abbildung 38 zeigt noch einmal den engen Zusammenhang zwischen der Access-Tabelle „Datenstruktur“ und der Informationstabelle auf.

Netzwerk

Die Methoden der SQL-Schnittstelle sind im eM-Plant Netzwerk .SQL.SQL (Abbildung 39) implementiert. Entsprechend der Konventionen im Projekt sind öffentliche Methoden durch ein grünes „M“ gekennzeichnet. Öffentliche Methoden können vom Benutzer oder von anderen Bausteinen aufgerufen werden. Es ist darauf zu achten, dass öffentliche Methoden Parameter erwarten können. Diese müssen beim Aufruf mit übergeben werden. Im SQL-Baustein sind öffentliche Methoden zur Initialisierung der Datentabellen, Transfer einzelner Tabellen und zum Transfer ganzer Datenkomplexe¹ vorgesehen. Im folgenden wird nur auf die Beschreibung der öffentlich zugänglichen Methoden des SQL-Bausteins eingegangen, da diese die externe Schnittstelle des Bausteins darstellen.

- **Methoden zur Initialisierung**

- **ErzeugeInfo**

erforderliche Parameter : Datenkomplex, ODBC_DSN, Benutzer, Kennwort

Stellt über die ODBC-Benutzerdatenquelle eine Verbindung zur Datenbank her. Anschließend wird überprüft, ob zu jeder Datentabelle eine Informationstabelle vorhanden ist. Ist noch keine vorhanden, wird diese erzeugt. Darauf werden die Informationen zur Struktur, Schlüsseln und Access-Tabellenbezeichnung aus der Datenbanktabelle „Datenstruktur“ ausgelesen und die Informationstabelle entsprechend gefüllt.

¹ Datenkomplexe sind im Projekt entwickelte Objekte, die eine Sammlung zusammengehöriger Tabellen mit ihren Zugriffsmethoden repräsentieren.

Nach dem Ausführen der Methode sind die Datentabellen gelöscht. Eingestellte Wertebereich in den Infotabellen gehen verloren.

- **Methoden zur Bearbeitung einzelner Datentabellen**

Die Methoden zur Bearbeitung einzelner Datentabellen bestehen aus zwei Gruppen. Die eine ist zur Verwaltung der Tabellen zuständig, die andere führt die gewünschte Operation auf den Tabellen aus.

- **InitTable**

Keine Parameter erforderlich.

Stellt sicher, dass vor der Auswahl einzelner Tabellen keine andere Auswahl getroffen wurde.

- **SetTable / SetTableUser**

Erforderlicher Parameter: Zu sichernde Tabelle als Objekt

(zusätzlich bei SetTableUser: Benutzer und Kennwort)

Ermittelt aus der Informationsdatei den Namen der entsprechenden Access-Tabelle und die erforderliche ODBC-Benutzerdatenquelle. Sind die entsprechenden Werte ermittelt, werden diese als neue Zeile in die Tabelle „TabellenAuswahl“ eingetragen. Ist für den Zugriff ein Benutzer erforderlich, so sollte die Methode SetTableUser eingesetzt werden. Diese Methode entspricht in der Funktionalität der Methode SetTable, es kann zusätzlich noch ein Benutzer und ein Kennwort angegeben werden, um auf die Datenbank zuzugreifen.

- **RemoveTable**

Erforderlicher Parameter: Name der zu entfernenden Tabelle

Entfernt den Eintrag der Tabelle, die durch den als Parameter übergebenen Namen gekennzeichnet ist, aus der Auswahl Tabelle.

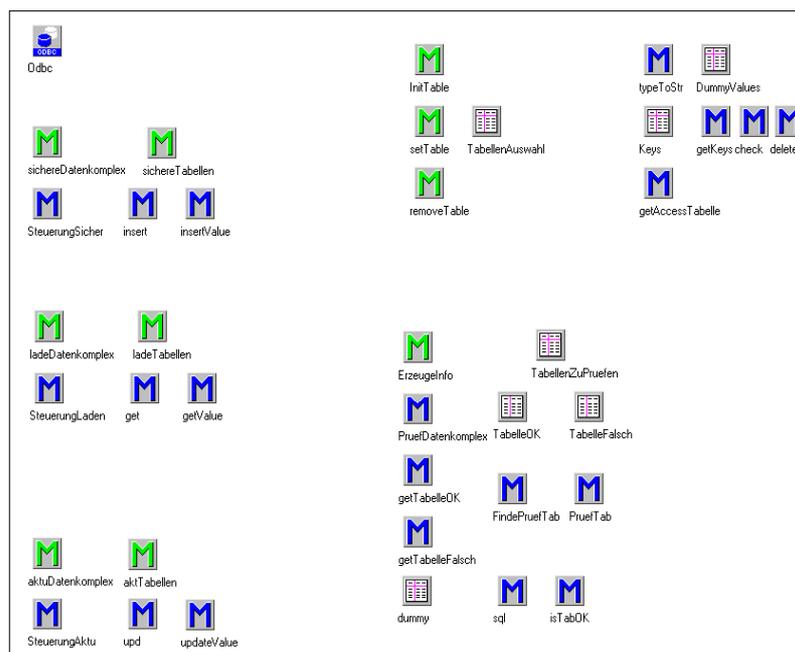


Abbildung 39 - SQL-Baustein: Netzwerk der SQL Schnittstelle

- **SichereTabellen**

Keine Parameter erforderlich. Alle benötigten Werte sind durch die Methode „Set-Table“ ermittelt worden.

Schreibt die Inhalte der ausgewählten Datentabellen in die Access-Datenbank. Entspricht das Satzformat nicht den von Access geforderten Format, wird die Aktion auf der entsprechenden Tabelle nicht ausgeführt.
- **LadeTabellen**

Keine Parameter erforderlich. Alle benötigten Werte sind durch die Methode „Set-Table“ ermittelt worden.

Lädt die Daten der gewählten Tabellen aus der Datenbank. Hierbei werden die Wertebereiche der Informationstabelle berücksichtigt. Entspricht das Satzformat nicht dem geforderten Format, wird die Informationstabelle und die Datentabelle entsprechend dem Datenformat in der Datenbank angepasst. Datenfelder, die Schlüsselfelder sind, werden in der Datentabelle auf schreibgeschützt gesetzt.
- **AktuTabellen**

Keine Parameter erforderlich. Alle benötigten Werte sind durch die Methode „Set-Table“ ermittelt worden.

Aktualisiert die Inhalte der ausgewählten Datentabellen in die Access-Datenbank. Entspricht das Satzformat nicht dem von Access geforderten Format, wird die Aktion auf der entsprechenden Tabelle nicht ausgeführt.
- **Methoden zur Bearbeitung von Datenkomplexen**
 - **SichereDatenkomplex**

Erforderliche Parameter : Datenkomplex, ODBC_DSN, Benutzer, Kennwort

Stellt eine Verbindung mit der Datenbank her. Prüft, ob die Struktur aller Tabellen im ausgewähltem Datenkomplex mit der Datenbank übereinstimmt. Sichert im Falle der Übereinstimmung aller Tabellen die Daten in der Datenbank.
 - **LadeDatenkomplex**

Erforderliche Parameter : Datenkomplex, ODBC_DSN, Benutzer, Kennwort

Stellt die Verbindung zur Datenbank her. Aktualisiert die Struktur der zu ladenden Tabellen und lädt anschließend die Daten. Datenfelder, die Schlüsselfelder sind, werden in der Datentabelle auf schreibgeschützt gesetzt.
 - **AktuDatenkomplex**

Erforderliche Parameter : Datenkomplex, ODBC_DSN, Benutzer, Kennwort

Stellt eine Verbindung mit der Datenbank her. Prüft, ob die Struktur aller Tabellen im ausgewähltem Datenkomplex mit der Datenbank übereinstimmt. Aktualisiert im Falle der Übereinstimmung aller Tabellen die Daten in der Datenbank.

2.2.4 Toolset Schiffbauliche Fertigung

Das Toolset Schiffbauliche Fertigung² ermöglicht den effizienten Aufbau von neuen und einfache Änderungen an bestehenden Modellen. Aufgrund der sehr allgemein gehaltenen Struktur vor allem der Steuerungskomponenten ist es auch möglich, fertigungsfremde Werftbereiche wie Konstruktionsbüros mit Hilfe des Toolsets zu simulieren.

In diesem Abschnitt werden vorweg die Hauptvorteile des Toolsets genannt. Es folgen Beschreibungen der vorhandenen Komponenten. Dabei wird auch der Werdegang des Toolsets - von einem zentralen Leitrechner mit komplexer Funktionalität hin zu abgeschlossenen Komponenten mit verteilten Aufgaben - aufgezeigt. Abschließend wird kurz auf die Vorgehensweise bei der Erstellung von Modellen und Ressourcenklassen mit Hilfe des Toolsets eingegangen.

2.2.4.1 Möglichkeiten und Anwendungsvorteile des Toolsets

Mit dem Simulationstoolset können komplexe Fertigungsumgebungen einfach und in kurzer Zeit auf- und flexibel umgebaut werden. Dazu sind nur wenige Schritte notwendig, die meisten Parameter werden im Datenbaustein hinterlegt. Das Toolset vereint folgende Eigenschaften:

- Hoher Funktionsumfang
z.B. Abbildung von Arbeitsplänen, Prozessen, Qualifikationen, Ressourcen und detailliertem Materialfluss mit weitreichenden Parametrisierungsmöglichkeiten
- Objektorientierte Struktur
Durch die offene, objektorientierte und klar gegliederte Struktur kann der erfahrene Anwender Änderungen oder Erweiterungen einfach und konsistent in das Toolset implementieren. Die Steuerungsaufgaben sind, in logische Gruppen gegliedert, auf zentrale oder lokale Manager verteilt. Im Sinne einer möglichst weitgehenden Objektorientierung werden dabei Daten und Funktionen direkt den Objekten zugeordnet, soweit sie diese betreffen.
- Einheitliche Schnittstellen
Die Kommunikation der Toolsetobjekte findet über öffentliche Schnittstellenmethoden und standardisierte Meldungen und Datenpakete statt. Es wird weder lesend noch schreibend direkt auf die Elemente (Methoden, Materialflussbausteine, BEs) der Objekte zugegriffen.
- Modulkonzept
An das Toolset sind verschiedene gekapselte Module, z.B. der Data-, Daten-, Koordinaten-, Prozesszeiten-, SIMPLERMethoden- und Pausenbaustein gekoppelt. Die Bausteine können getrennt voneinander bearbeitet und konfiguriert werden.
- Eventverarbeitung

² Das Toolset Schiffbauliche Fertigung wurde gemeinsam mit der Firma SIPOC GmbH entwickelt. Dieser Abschnitt enthält Auszüge aus der von SIPOC erstellten ausführlichen Dokumentation des Toolsets.

Alle relevanten Simulationsereignisse können der Eventschnittstelle übergeben werden. Die Informationen können zur Visualisierung oder für Protokollfunktionen verwendet werden.

- Visualisierung

Die Verfah- und Bearbeitungsvorgänge z.B. der beweglichen Maschinen werden zur Laufzeit dargestellt.

2.2.4.2 Manager auf Ressourcennetzwerk- und Ressourcenebene

In diesem Abschnitt werden diejenigen allgemeinen Steuerungskomponenten vorgestellt, die in Simulationsmodellen an unterschiedlichen Orten zu instanzieren sind. Eine zentrale Rolle spielt unter diesen Bausteinen der Simulationsleitreechner, der das Zusammenspiel der für Teilfunktionen der Simulationssteuerung spezialisierten Manager koordiniert. Bevor die Komponenten im einzelnen beschrieben werden, wird zunächst ein kürzer Überblick über die Entwicklungsstufen des Leitreechnerkonzepts im Laufe des Projekts gegeben.

1. Bausteinkasten „Schiffbauliche Vorfertigung“

Die erste Phase war die Entwicklung eines „Toolsets für die schiffbauliche Vorfertigung“ mit dem Ziel, stahlbautypische Arbeitsgänge abzubilden (Bearbeitung, Montage, Demontage). Im Vordergrund stand das Modellieren der Ressourcenlogik (Materialfluss). Gleichzeitig wurden die Möglichkeiten des neu erworbenen Simulationssystems eM-Plant ausgelotet.

Dabei wurde bald erkannt, dass für die logische Abbildung der Werftabläufe allgemein zunächst ein umfassenderes Konzept verfolgt werden musste. Dies betraf die Definition einer allgemeinen Datenstruktur (2.1), zum anderen die Entwicklung allgemeiner Module, die unabhängig von ressourcenspezifischen Regeln den Ablauf steuern.

2. Zentraler Simulationsleitreechner

Die zweite Phase war von der Entwicklung eines allgemeinen Simulationsleitreechners bestimmt. Diese verlief parallel zur Definition der allgemeinen Simulationsdatenstruktur. Das Resultat war ein Baustein, der sämtliche Steuerungsfunktionen in sich vereinigte. Ein Großteil davon war in einem separaten Netzwerk, dem Bauteilrepräsentanten, implementiert (Abbildung 40). Für jedes simulierte Bauteil wurde zur Simulationslaufzeit eine Instanz des Bauteilrepräsentanten erzeugt, in der relevante Bauteildaten, aber auch Steuerungslogiken gespeichert waren. Diese Entscheidung wurde auch deswegen getroffen, weil in der damals aktuellen Version eM-Plant 4.6 Bauteilen keine Steuerungslogiken in Form von Methoden attribuiert werden konnten.

Nachteilig war neben der Performanceverluste durch das Instanzieren und Initialisieren von Netzwerken zur Simulationslaufzeit, dass die Ausführung der unterschiedlichen Teilfunktionen (nächsten Arbeitsschritts bestimmen, Ressourcen zuordnen, Bauteiltransport planen etc.) - teils innerhalb des Leitreechners, teils in Kommunikation mit den beteiligten Ressourcen - auf eine äußerst verschachtelte Weise programmiert und mit z.T. tiefen Methodenaufufrufketten verbunden war. Die Einarbeitung auch geübter Entwickler war entsprechend schwierig und der Leitreechner kaum wartungsfähig.

3. Komponenten „Schiffbauliche Fertigung“

Es wurde erkannt, dass die im Leitreechner implementierten Teilfunktionen klar gegliedert und dann in einzelne Komponenten aufgeteilt werden mussten, die schließlich im Simulationsablauf möglichst nacheinander statt ineinander verschachtelt auszuführen sind. Diese Aufgabe

prägte die dritte Phase, in der die Funktion des Leitrechners selbst auf das schrittweise Anstoßen der in Netzplänen festgelegten Arbeitsschritte und die Aktivierung der zu deren Ausführung benötigten Komponenten reduziert wurde. Der Bauteilrepräsentant wurde ganz aus dem Toolset entfernt, bauteilrelevante Daten werden nunmehr an den BEs (die eigentlichen Bauteilrepräsentanten, die auch Platz auf den Ressourcen belegen) gespeichert.

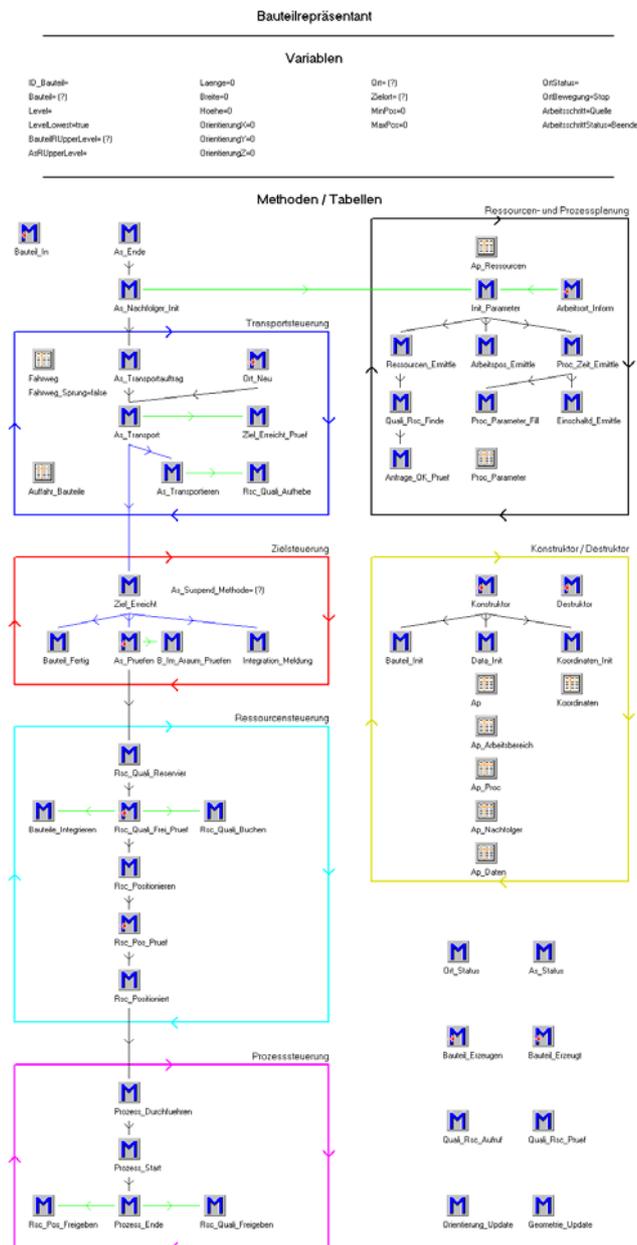


Abbildung 40 - Manager auf Ressourcennetzwerk- und Ressourcenebene: Bauteilrepräsentant mit Leit-rechnerfunktionen

2.2.4.2.1 Leitrechner



Der Leitrechner ist die zentrale Steuerungsinstanz des Toolsets und wird in das Hauptnetzwerk jedes Modells eingesetzt. Seine Aufgabe ist es, auf Anfrage eines Arbeitsobjekts (Bauteil, Zeichnung o.a.) dessen nächste(n) im Netzplan eingetragene(n) Arbeitsschritt(e) zu ermitteln und dann im Zusammenspiel mit den unterschiedlichen Leit-rechnerkomponenten und Ressourcen die Ausführung einzuleiten.

G:\Abschluss\Bericht\Gesamtbericht v03.11.doc

Interne Abläufe

Über die öffentliche Methode „Next_As“ wird die Abarbeitung von Arbeitsschritten eingeleitet. Die damit eingeleitete Arbeitsschrittabarbeitung führt unter Verwendung mehrerer externer Steuerungskomponenten verschiedene Planungsschritte nacheinander durch:

- Identifizierung des nächsten Arbeitsschritts

Die Methode „Next_As“ lässt den Arbeitsplanmanager den folgenden Arbeitsschritt identifizieren. Die Arbeitsschrittschlüssel werden am BE gespeichert.

Sind auf der aktuellen Arbeitsplanebene alle Arbeitsschritte abgearbeitet, so wird auf der nächst höheren Ebene ein nachfolgender Arbeitsschritt gesucht.

Es werden sequentielle und alternative Arbeitsschrittverknüpfungen unterstützt. Die Alternativenauswahl findet über im SIMPLEMethoden-Netzwerk hinterlegte Entscheidungsmethoden statt.

- Ermittlung der möglichen Ressourcenpakete

Die von „Next_As“ im Anschluss aufgerufene Methode „RscPakete_Find“ ermittelt über die Komponente Ressourcenfinder die zur Ausführung des Arbeitsschritts in Frage kommenden Ressourcenpakete. Sie werden als Attribut in Tabellenform am BE gespeichert.

- Ermittlung der möglichen Transportwege

Die Methode „Transportweg_Find“ ermittelt für jedes Ressourcenpaket den (optimalen) Transportweg vom aktuellen Aufenthaltsort des BEs zum jeweiligen Arbeitsort. Sie werden als Attribut in Tabellenform am BE gespeichert.

- Belastungsermittlung der Pakete

Für alle Pakete wird über Aufruf der Belastungsprüfung von der Methode „RscPakete_Get_Belast“ die summierte zeitliche Belastung der im Paket enthaltenen Ressourcen evaluiert. Sie werden als Attribut in Tabellenform am BE gespeichert.

- Auswahl eines Ressourcenpakets

Die Methode „RscPakete_Select“ wählt ein Ressourcenpaket zur Ausführung des Arbeitsschritts aus. Sie richtet sich dabei nach der Belastung der in den Paketen enthaltenen Ressourcen. Die alternativen Pakete werden nicht aus den Tabellen gelöscht. Stattdessen wird ein neues Attribut an das BE gefügt, welches einen Zeiger auf das gewählte Paket enthält.

- BE Transport

Nachdem alle notwendigen Informationen zur Durchführung des Arbeitsschritts zur Verfügung stehen, wird dem aktuell zuständigen Transportmanager der Befehl zum Transport des BEs gegeben.

2.2.4.2.2 Transportmanager



Der Transportmanager dient sowohl als BE- oder Materialflussschnittstelle eines Netzwerks zur weiteren Simulationsumgebung, als auch der Steuerung des Transports zwischen den Ressourcen eines Netzwerks. In jedem Fertigungsnetzwerk muss ein Transportmanager instanziiert werden.

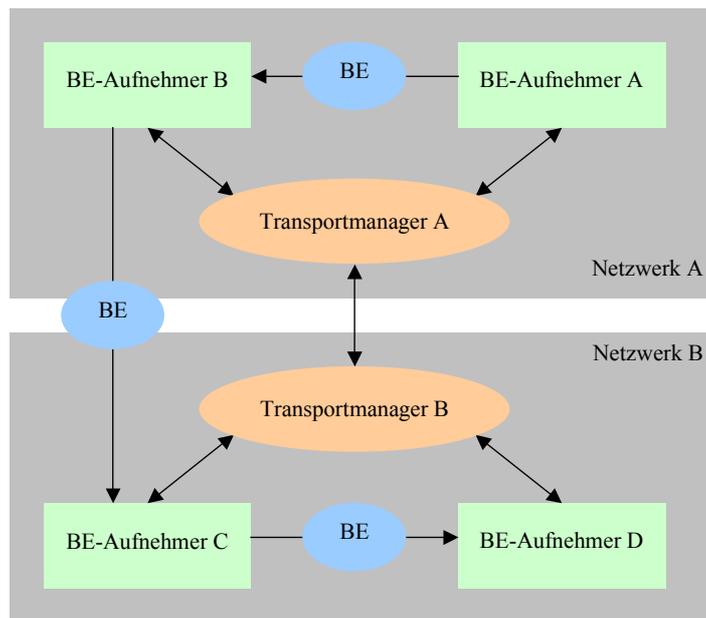


Abbildung 41 - Transportmanager: Funktionsübersicht

Es handelt sich bei dem hier erläuterten Element um den Standardtransportmanager zur Verwaltung von anzahlorientierten BE-Aufnehmern. Dazu gehören:

- Quelle
- Senke
- Container

Für andere Arten des BE-Handling werden angepasste Transportmanager benötigt. So existiert für die längenorientierten Ressourcen der Paneelstraße ein eigener Transportmanager. Die Schnittstellen zu den BE-aufnehmenden Ressourcen des verwalteten Netzwerks sind dementsprechend angepasst. Die Schnittstellen zu den externen Managern, dem Leitreechner und auch dem internen Jobmanager entsprechen jedoch, um die Kompatibilität zu wahren, dem Standard.

2.2.4.2.3 Jobmanager



Der Jobmanager dient der Ressourcen- und Prozessverwaltung. Hat ein BE seinen Arbeitsort erreicht, übernimmt der Jobmanager vom lokalen Transportmanager den Steuerungsfokus. Er koordiniert die genaue Positionierung des BEs und der Arbeitsressourcen. Anschließend lastet er die zu verwendenden Ressourcen ein und verwaltet den Arbeitsprozess. Unterbrechungen (Störungen, Pausen etc.) einzelner Ressourcen werden an den Jobmanager gemeldet, woraufhin er die Beendigung des Prozesses verzögert.

Ein Jobmanager ist in jedes Ressourcennetzwerk einzusetzen.

2.2.4.2.4 Ressourcenmanager



Der Ressourcenmanager verwaltet das Arbeitsangebot arbeitsleistender Ressourcen. Alle arbeitsleistenden Ressourcen benötigen einen Ressourcenmanager oder müssen in der Lage sein, auf die für arbeitsleistende Ressourcen definierten Meldungen reagieren können. Die Verwendung einer Ressource durchläuft in der Regel fünf Phasen:

1 Anfrage

Über die Anfrage wird der Ressource gemeldet, dass sie in naher Zukunft verwendet werden wird. Dem Ressourcenmanager einer Ressource zunächst eine Anfrage zu melden ist z.B. sinnvoll, falls die Ressource für einen Prozess ausgewählt wurde, das BE aber erst noch in den Arbeitsraum der Ressource transportiert werden muss. Die Anfrage dient einer genaueren Belastungsermittlung der Ressource und muss nicht unbedingt vor einer Verwendung durchgeführt werden.

2 Reservierung

Mit "Reservieren" wird der Ressource gemeldet, dass sie gebucht werden soll, sobald sie frei wird. Wurde vor der Reservierung eine Anfrage für die betreffende Auftragsbezeichnung ausgeführt, so wird diese aufgehoben. Wird die (anzahlverwaltende Ressourcen: „eine“) Ressource frei, so wird die Liste der Reservierungen nach Priorität und Reservierungszeitpunkt sortiert und die Ressource den in der Liste enthaltenen Managern angeboten. Die Ressourcenreservierung vor einer Verwendung ist optional.

3 Buchung

Nachdem geprüft wurde ob die Ressource frei ist (z.B. über „Status_Ermitteln“ oder die Prüffunktionen), kann ihr die Meldung "Buchen" übermittelt werden. Wurde vor der Buchung eine Anfrage oder Reservierung ausgeführt, so wird diese aufgehoben. Eine Buchung belegt die Ressource, sie wird aber noch nicht genutzt. Die Ressourcenbuchung vor einer Verwendung ist optional.

4 Nutzung

Wurde eine Buchung durchgeführt oder geprüft ob die Ressource frei ist, so kann die Meldung "Nutzen" übermittelt werden. Sie zeigt an, dass die Ressource ab diesem Zeitpunkt tatsächlich verwendet wird.

5 Freigabe

Die Meldung "Freigeben" beendet die Nutzung einer Ressource. Liegen weitere Reservierungen vor, so wird dies dem Jobmanager angezeigt.

2.2.4.3 Allgemeine Steuerungskomponenten

Einige Komponenten des Toolsets müssen nicht ein oder mehrmals in Modellen eingesetzt werden. Es genügt das Vorhandensein der Klasse. Diese Komponenten werden im folgenden behandelt.

2.2.4.3.1 Arbeitsplanmanager



Der Arbeitsplanmanager erlaubt den Zugriff auf die arbeitsschrittbezogenen Information des Datenbausteins. Dazu gehört die Ermittlung und Rückgabe von Nachfolger- und Vorgängerarbeitsschritten, sowie die Navigation zwischen den Arbeitsplanebenen. Die gewonnenen Informationen werden der anfragenden Komponente – z.B. dem Leiterrechner – in strukturierten Tabellen geliefert.

Öffentliche Methoden

- **Get_Next_As, Get_Prev_As**

Diesen Methoden wird als Parameter ein Arbeitsschritt übergeben. Sie werten die Ap_Nachfolger-Tabelle des Datenbausteins aus und geben alle im Netzplan definierten

direkten Nachfolge- bzw. Vorgängerarbeitsschritte zurück (Datensätze der Ap_Nachfolger-Tabelle, ohne Schlüssel).

- **Get_Lower_As**

Mit dieser Methode werden zu einem gegebenen Arbeitsschritt alle hierarchisch untergeordneten Arbeitsschritte ermittelt (Datensätze der Ap-Tabelle, ohne Schlüssel).

- **Get_Lower_First, Get_Lower_Last**

Diese Methoden prüfen für einen angegebenen Arbeitsschritt, ob es hierarchisch untergeordnete Arbeitsschritte gibt, und geben den ersten bzw. letzten Arbeitsschritt des untergeordneten Netzplans zurück (Datensatz der Ap_Nachfolger-Tabelle, ohne Schlüssel).

- **Get_Upper_As**

Zu einem angegebenen Arbeitsschritt liefert diese Methode den vollständigen Datensatz zum hierarchisch übergeordneten Arbeitsschritt (Datensatz der Ap -Tabelle).

2.2.4.3.2 Ressourcenfinder



Der Ressourcenfinder enthält Methoden zum Bestimmen möglicher Ressourcen zur Durchführung eines Arbeitsschritts. Die im Datenbaustein gespeicherten Informationen werden ausgewertet, um geeignete Bauteilaufnehmer und Qualifikationsanbieter zu ermitteln, bereits getroffene Vorauswahlen bezüglich des Arbeitsortes zu berücksichtigen etc.

Öffentliche Methoden

- **Rsc_Finde**

Die Methode gibt zum angegebenen Arbeitsschritt die benötigten Qualifikationen und in Untertabellen die in Frage kommenden Arbeitsressourcen zurück. Falls in Ap_AS_Rsc eine Auswahl des Arbeitsortes gefunden wurde, entfallen die ortsfernen Arbeitsressourcen.

- **Rsc_Finde_Multi_Get**

Der Methode wird eine Tabelle mit Arbeitsschritten übergeben. Die Tabelle wird durchlaufen und die Methode Rsc_Finde iterativ aufgerufen. Die erhaltenen Rückgabewerte werden in Untertabellen der an Rsc_Finde_Multi_Get übergebenen Tabelle eingetragen.

- **Rsc_Finde_Angebot**

Der Methode wird ein Arbeitsschritt übergeben. Die Rückgabetable enthält je angeforderter Qualifikation und potentiellm Arbeitsort die möglichen Arbeitsressourcen inklusive Anzahl verfügbarer Ressourceneinheiten. Qualifikationsanbieter ohne Ortszuordnung werden in die letzte Zeile ("Beliebig") geschrieben. Gegebenenfalls sind mehr geeignete Qualifikationsanbieter vorhanden als erforderlich, so dass sich eine Vielzahl an Möglichkeiten ergibt, den Prozess mit unterschiedlicher „Besetzung“ auszuführen.

- **Rsc_Finde_Pakete**

Der Methode wird ein Arbeitsschritt übergeben, dann wird die Methode Rsc_Finde_Angebot ausgeführt. Nachdem so die möglichen Arbeitsorte und das dort jeweils bestehende Angebot an Arbeitsressourcen ermittelt wurde, werden sämtliche möglichen Ressourcenpakete ermittelt.

2.2.4.3.3 Transportplaner



Der Transportplaner generiert Transportwege anhand der im Datenbaustein hinterlegten Standort- und Verknüpfungsdaten. Transportweglängen werden beachtet, um einen möglichst kurzen Weg zu finden.

Öffentliche Methoden

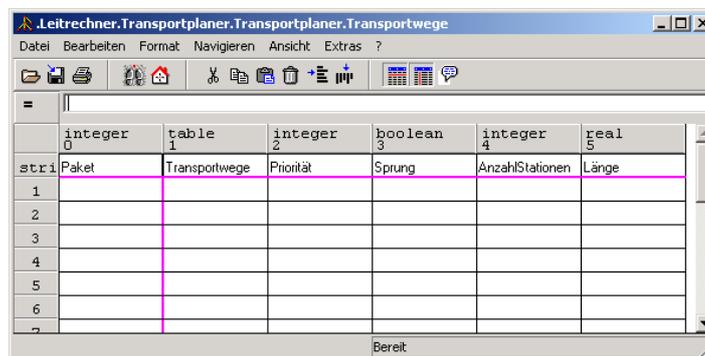
- **Get_Transportwege**

Die Methode versucht für jedes Paket den optimalen Transportweg anhand der zurückzulegenden Strecken zu ermitteln. Ihr werden dazu folgende Informationen übergeben:

- aktueller Standort des BEs
- Position des BEs
- durch den Ressourcenfinder ermittelte Ressourcenpakete
- Tabelle der Modellressourcen

Die gefundenen Transportwege werden in einer Tabelle zurückgegeben (siehe Abbildung 42). Sie enthält folgende Einträge:

- Referenznummer des Ressourcenpakets
- Transportweguntertabellen

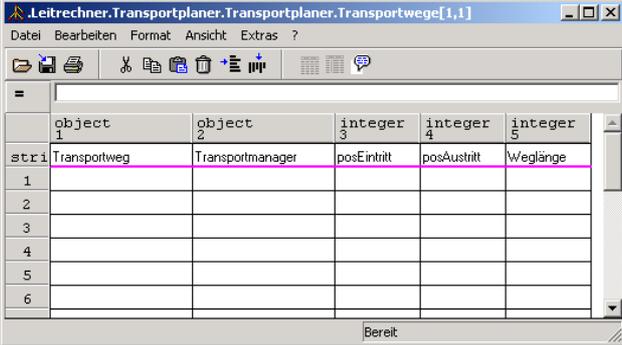


	integer 0	table 1	integer 2	boolean 3	integer 4	real 5
str1	Paket	Transportwege	Priorität	Sprung	AnzahlStationen	Länge
1						
2						
3						
4						
5						
6						
7						

Abbildung 42 - Get_Transportwege Rückgabetable

Die in Abbildung 43 dargestellten Untertabellen enthalten die Transportstationen, die jeweils zuständigen Transportmanager, die Ein- und Austrittsposition und die zurückgelegte Weglänge

- Priorität des Pakets aus Sicht des Transportplaners
- „Sprung“-Angabe (Angabe, ob eine BE-Übergabe zwischen zwei Ressourcen, die nicht verknüpft sind, im Transportweg enthalten ist)
- Anzahl der im Transportweg enthaltenen Stationen
- Gesamtlänge des Transportwegs



	object 1	object 2	integer 3	integer 4	integer 5
stri	Transportweg	Transportmanager	posEintritt	posAustritt	Weglänge
1					
2					
3					
4					
5					
6					

Abbildung 43 - Get_Transportwege Untertabelle Transportwege

2.2.4.3.4 Belastungsprüfung

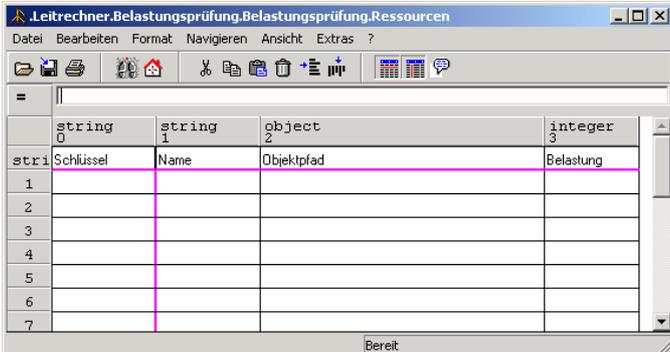


Die Belastungsprüfung dient der Erfassung der Belastungssituation von Ressourcen. Sie ermittelt für die mit dem Ressourcenfinder identifizierten Pakete die jeweilige zeitliche Gesamtbelastung der angegebenen Ressourcen.

Öffentliche Methoden

- **Get_Rsc_Belastung**

Die Methode ermittelt die zeitliche Belastung von Ressourcen. Ihr werden die Ressourcenpakete und eine Tabelle der im Simulationsmodell enthaltenen Ressourcen übergeben. Sie fragt alle zuständigen Jobmanager der in den Paketen enthaltenen Ressourcen an. Diese geben die Statusabfrage an die jeweilige Ressource weiter und melden die Belastung zurück an die Belastungsprüfung. Diese gibt eine Tabelle mit der in Abbildung 44 dargestellten Struktur zurück.



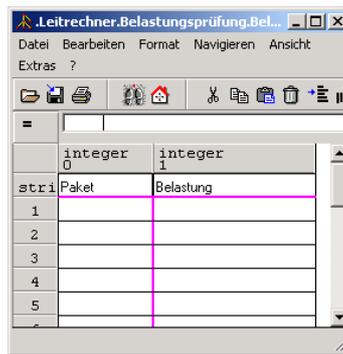
	string 0	string 1	object 2	integer 3
stri	Schlüssel	Name	Objektpfad	Belastung
1				
2				
3				
4				
5				
6				
7				

Abbildung 44 - Get_Rsc_Belastung Rückgabetable

Sie enthält einen Schlüssel, zusammengesetzt aus Ressourcenklasse und -name, den Ressourcennamen, den Objektpfad zur Ressource und ihre zeitliche Gesamtbelastung (Ermittlung siehe Ressourcenmanager).

- **Get_Pakete_Belastung**

Die Methode unterscheidet sich von der Methode Get_Rsc_Belastung nur im Format ihrer Rückgabetable. Sie gibt nicht die Belastung pro Ressource sondern pro Ressourcenpaket zurück (siehe Abbildung 45). Für jedes Paket wird die summierte Belastung der Ressourcen zurückgeben.



	integer 0	integer 1
str1	Paket	Belastung
1		
2		
3		
4		
5		

Abbildung 45 - Get_Pakete_Belastung Rückgabetable

2.2.4.3.5 Prozesszeitmanager



Der Prozesszeitmanager dient der Bestimmung der für einen Arbeitsschritt einzuplanenden Prozesszeit. Die am Prozess teilnehmenden Ressourcen müssen zuvor, z.B. mit dem Baustein „Ressourcenfinder“, ermittelt worden sein. Weiterhin ermittelt er die auf den Transportvorgang aufzuschlagenden Zeiten für den direkten Wechsel eines Bauteils zwischen zwei Standorten (Übergangszeiten).

Öffentliche Methoden

- **Get_Prozesszeit**

Der Methode wird der vollständige Schlüssel des auszuwertenden Arbeitsschritts, die Tabelle mit den nutzbaren Ressourcenkombinationen (Ressourcenpakete) und die Nummer der auszuwertenden Ressourcenkombination übergeben. Sie ermittelt die einzuplanende Prozesszeit für den Auftrag.

Nur wenn für den Arbeitsschritt in der Ap-Tabelle keine "ZeitdauerStandard" größer 0 angegeben ist, werden die ressourcen- und arbeitsplanbedingten Attribute ermittelt und von einer speziellen Methode ausgewertet. Ist eine -1 hinterlegt, so wird dies als undefinierte Prozesszeit gewertet. Diese Prozesse werden durch ein "externes" Ereignis beendet (z.B. Prozess "Einlagern" => Ende, wenn von übergeordneter Baugruppe die Auslagerung angefordert wird).

- **Get_Übergangszeit**

Unter Umständen müssen bei einem längeren Bauteiltransport zusätzliche Zeiten beim Wechsel des Bauteilstandorts beachtet werden, die z.B. aus dem Handling bei der Übergabe des Bauteils resultieren. Die Methode Get_Übergangszeit ermittelt diese Zeiten. Der Methode werden Quell- und Zielressource, Topologie und Sicht des betrachteten Bauteils übergeben.

2.2.4.3.6 Koordinaten-Komponente



Die Koordinaten-Komponente dient der Ermittlung von Bauteilabmessungen aus der Produktstruktur. Mit einer Reihe öffentlicher Methoden werden die notwendigen Koordinatentransformationen abgedeckt, die im Laufe der Fertigungsprozesse anfallen (Drehungen in 90-Grad-Schritten um alle Raumachsen, Änderungen der Abmessungen im Laufe des Zusammenbaus). Hier werden nur die beiden im Zusammenspiel mit dem Toolset gebräuchlichsten Methoden genannt:

Öffentliche Methoden

- **IntegrationAuswerten**

Die Methode wertet Integrationsarbeitsschritte aus. Es werden dabei drei Fälle unterschieden:

1. Bauteil wird auf Quelle erzeugt (Eingangsbauteil nicht definiert)
2. Bauteil entsteht durch Integration seines ersten Eingangsbauteils
3. Integration eines Eingangsbauteils zum bereits bestehenden Bauteil

Ihr werden die ID des Bauteils und des Eingangsbauteils, die absoluten Koordinaten der BoundingBox des Bauteils seit dem letzten Integrationsschritt und der Objektpfad der Produktstruktur übergeben.

Die Rückgabe ist eine einzeilige Tabelle

- 1.-24. Spalte: Neue Koordinaten der Bauteilumhüllenden (BoundingBox)
- 25.-27. Spalte: Neue Bauteilabmessungen in X,Y und Z-Richtung des Schiffs
- 28.-33. Spalte: Lage des Arbeitspunktes für die möglichen Orientierungen des Bauteiles zum Platz. Der Arbeitspunkt wird als nichtnegativer Wert der in Platzrichtung weisenden Bauteilkoordinate, bezogen auf die hinterste Kante des Bauteils, angegeben

- **Transform_XYZ**

„Transform_XYZ“ führt eine Koordinatentransformation vom Ursprungs- ins Zielkoordinatensystem durch. Objektdrehungen müssen immer 0 oder ein (negatives) vielfaches von 90 Grad in der Reihenfolge X, Y, Z bezogen auf das Weltkoordinatensystem sein.

Der Methode werden die zu transformierenden Koordinaten, das Ursprungskoordinatensystem und das Zielkoordinatensystem als dreizeilige Vektoren in Tabellenform übergeben. Sie liefert die transformierten Koordinaten in gleicher Form zurück.

2.2.4.3.7 SIMPLEMethoden



In diesem Baustein werden Spezialmethoden modelliert, die diverse Sonderfälle für die Simulation abdecken. Es handelt sich z.T. lediglich um Gerüste, d.h. Netzwerke, in die vorformulierte Methoden eingesetzt und mit bestimmten Logiken programmiert werden können.

Öffentliche Methoden

Alle Methoden des Bausteins sind öffentliche Methoden. Der Baustein ist in drei Sektionen gegliedert:

- Die Methoden der *Ablaufsuspendierung* werden aktiviert, wenn während der Arbeitsplanbearbeitung in der Spalte "ID_Bedingung_N" der Tabelle "Ap_Nachfolger" der Eintrag "CHECK" gefunden wurde. Sie können eingesetzt werden um die Abarbeitung von Arbeitsschritten direkt nach ihrer Einlastung durch die Methode „Next_As“ des Leitrechners zu unterbrechen.
- "START": Diese Methoden werden aktiviert, wenn in der Arbeitsplanbearbeitung in der Spalte "ID_Bedingung_N" der Tabelle "Ap_Nachfolger" der Eintrag "START" gefunden wurde (nicht im Toolset implementiert). Die Methoden können die Wiederaufnahme suspendierter Arbeitsplanbearbeitungen einleiten.

- **"ODER"**: Diese Methoden werden aktiviert, wenn in der Arbeitsplanbearbeitung in der Spalte "ID_Bedingung_N" der Tabelle "Ap_Nachfolger" der Eintrag "ODER" gefunden wurde (nicht im Toolset implementiert). Die Methoden finden bei alternativen Arbeitsschritten eine Entscheidung und tragen den Schlüssel des nächsten auszuführenden Arbeitsschritts in die ersten drei "string"-Zellen des übergebenen Datenpakets ein.

2.2.4.4 Ressourcenklassen und ihre Elemente

Über die Standardressourcen (BE-aufnehmende, arbeitsleistende oder hybride Ressourcen) wird die Produktionsumgebung abgebildet. Abbildung 46 zeigt die Gliederung der Ressourcen in Arbeitsanbieter, BE-Aufnehmer und Hybridressourcen. Der jeweilige Ressourcentyp wird durch die von der Ressource angebotenen öffentlichen Methoden bzw. verarbeitbaren Meldungen definiert.

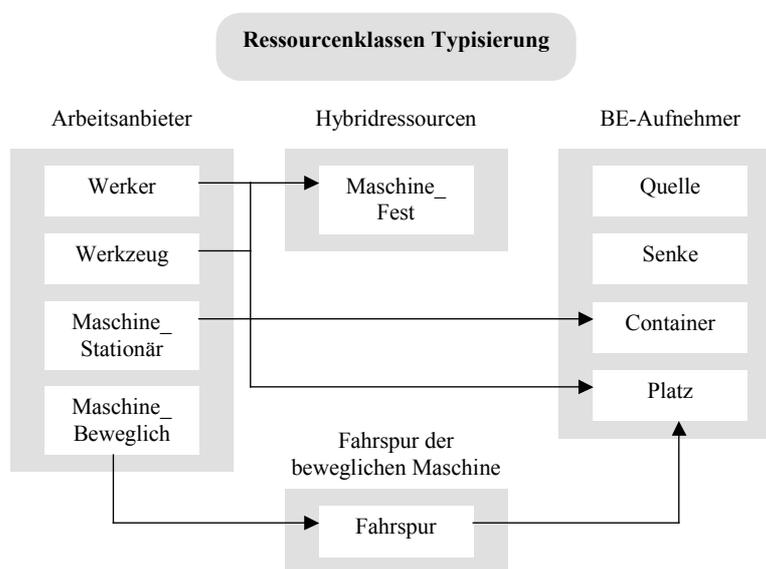


Abbildung 46 - Ressourcenklassen: Typisierung

Die in der Abbildung dargestellten Pfeile zeigen die möglichen Zuordnungen zwischen den Arbeitsanbietern und den BE-Aufnehmern.

So können z.B. der Werker, das Werkzeug und die stationäre Maschine BEs bearbeiten, die sich in Containern, auf Plätzen oder auf der „Maschine_Fest“ befinden. Die bewegliche Maschine befindet sich immer auf einer Fahrspur, die wiederum einem Platz zugeordnet ist. Auf der Quelle und der Senke können keine BEs bearbeitet werden. Die „Maschine_Fest“ ist eine hybride Ressource, die BEs aufnimmt und Arbeit leistet.

2.2.4.5 Ressourcenklassen

2.2.4.5.1 Ressourcenvorlagen

Das Toolset lässt sich um neue Ressourcenklassen erweitern. Die Ressourcen setzen sich, je nach Funktionstyp, aus verschiedenen Standardelementen zusammen. Die eingesetzten Elemente (bzw. die damit zur Verfügung stehenden Schnittstellen) definieren den Typ der Ressource. Es können sowohl BE-Aufnehmer, Arbeitsanbieter, Hybridelemente als auch benutzerdefinierte Ressourcen entwickelt werden.

- Rsc_Vorlage
Die allgemeine Ressourcenvorlage „Rsc_Vorlage“ dient dem Aufbau neuer Ressourcentypen.
- Rsc_Vorlage_ArbAnb
Die Klasse „Rsc_Vorlage_ArbAnb“ dient zum Aufbau von arbeitsleistenden Ressourcen.
- Rsc_Vorlage_BEAufn
Die Klasse „Rsc_Vorlage_BEAufn“ wird zur Erstellung neuer BE-Aufnehmender Ressourcen eingesetzt.
- Rsc_Vorlage_Hybrid
Die Hybridressourcen-Vorlage ist eine Kombination aus BE-Aufnehmer und Arbeitsanbieter.

2.2.4.5.2 Quelle

Die Quelle ist der Erzeugungsort von BEs. Jedes Modell benötigt mindestens eine Quelle. Quellen können im Haupt- wie auch in den untergeordneten Fertigungsnetzwerken eingesetzt werden.

Struktur

Das Netzwerk enthält das eM-Plant-Standardelement „Quelle“ und eine zugehörige Lieferliste in der Reihenfolge, Zeitpunkt, Orientierung und Attribute der BE-Erzeugung hinterlegt werden. Über die frei zu programmierende Methode „Lieferliste_Init“ kann die Lieferliste noch vor dem Simulationsstart manuell neu initialisiert werden. Der Inhalt der Lieferliste bleibt über die Simulationsläufe hinaus erhalten. Als Attribute werden dem BE angehängt:

- ID_Topologie
- Sicht

Über die Methode „Geom_Attrib_Einfügen“ können dem BE die für die Paneelstraße benötigten geometrischen Attribute angehängt werden:

- Länge
- Breite
- Höhe
- OrientierungX
- OrientierungY
- OrientierungZ

Interne Abläufe

Hat die Quelle ein BE erzeugt, so wird die Ausgangssteuerung „Quelle_Ausgang“ aufgerufen. Diese erzeugt die Attribute „ID_Arbeitsschritt“ und „Ort“ und weist ihnen „Quelle“ und die Objektreferenz der Quelle zu. Anschließend wird die Methode „BE_Erzeugt“ des lokalen Transportmanagers aktiviert.

2.2.4.5.3 Senke

Die Senke dient der Aufnahme von BEs deren Arbeitspläne abgearbeitet wurden. Es wird zunächst pro Modell nur eine Senke auf oberster Fertigungsnetzwerkebene vorgesehen die direkt vom Leitrechner angesteuert wird.

Struktur

Das Netzwerk enthält das eM-Plant-Standardelement „Lager“ zur Speicherung der BEs.

2.2.4.5.4 Container

Der Container dient der anzahlorientierten Lagerung von BEs. Container können im Haupt- wie auch in den untergeordneten Fertigungsnetzwerken eingesetzt werden.

2.2.4.5.5 Werker

Der Werker stellt über seinen Ressourcenmanager Arbeitsleistung zur Verfügung. Werker können im Haupt- wie auch in den untergeordneten Fertigungsnetzwerken eingesetzt werden.

2.2.4.5.6 Werkzeug

Das Werkzeug stellt über seinen Ressourcenmanager Arbeitsleistung zur Verfügung. Es entspricht in Aufbau und Funktion dem Werker.

2.2.4.5.7 Maschine_Stationär

Die stationäre Maschine stellt über ihren Ressourcenmanager Arbeitsleistung zur Verfügung. Sie entspricht in Aufbau und Funktion weitgehend dem Werker, kann aber einer bestimmten Position an einem Platz zugewiesen werden (definiert über die Variable „Position“).

2.2.4.5.8 Maschine_Fest

Die Ressource „Maschine_Fest“ kann sowohl BEs aufnehmen als auch Arbeit leisten. Der Hybridbaustein kombiniert die Elemente und Funktionalitäten eines Containers und eines Werkers.

2.2.4.5.9 Paneelstraße

Die in Abbildung 47 dargestellten Klassen der Paneelstraße dienen der räumlichen (bzw. ebenen) Abbildung von Bauteilen auf einem linear fördernden Transportsystem. Die Klassen sind dem Bibliotheksordner Paneelstrasse untergliedert.

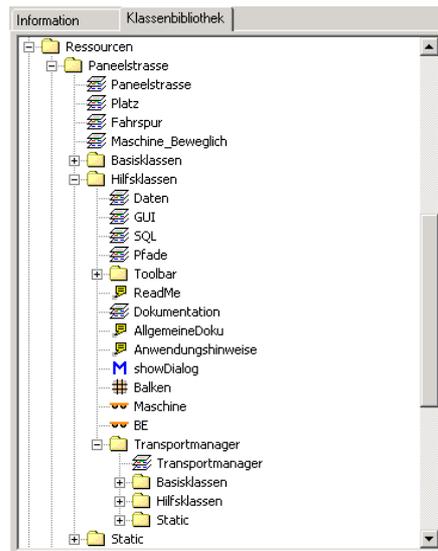


Abbildung 47 - Panelstraße: Klassenbibliothek

Abbildung 48 und Abbildung 49 zeigen das Panelstraßen-Ressourcennetzwerk. Das Netzwerk ist als Vorlage zum Aufbau von Panelstraßen-Fertigungsumgebungen konzipiert. Dazu kann es in das jeweilige Modell instanziiert und um weitere Panelstraßenklassen erweitert werden.

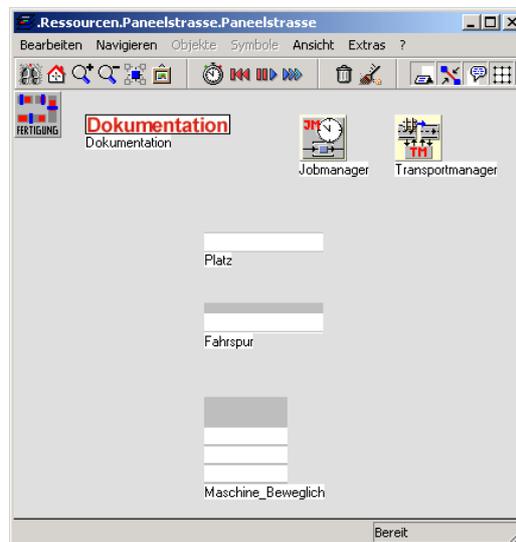


Abbildung 48 - Panelstraße: Layout des Ressourcennetzwerks

Das Netzwerk Panelstraße enthält folgende Elemente:

- Einen Standard-Jobmanager
- Einen auf die Schnittstellen der Panelstraßenklassen angepassten Transportmanager
- Einen Platz
- Eine Fahrspur
- Eine bewegliche Maschine

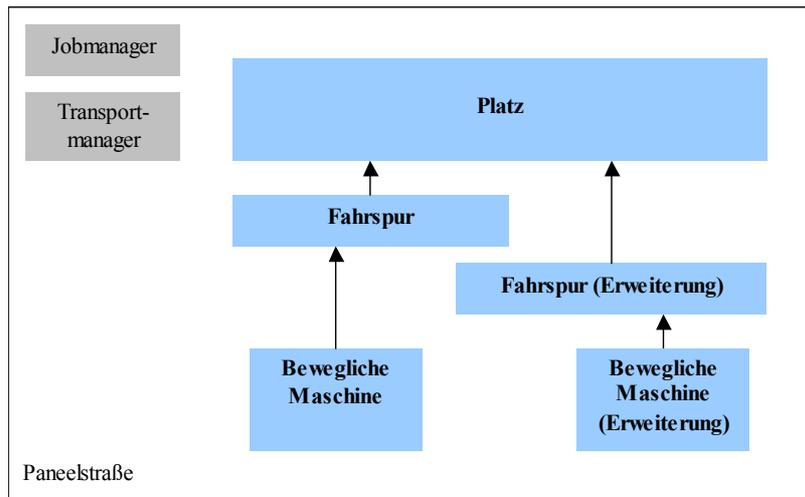


Abbildung 49 - Panelstraße: Erweiterung

2.2.4.5.10 Platz

Der in Abbildung 50 dargestellte Platz ist ein längenorientierter Aufnahme- und Bearbeitungsort von Bauteilen. Plätze können in Panelstraßen-Netzwerke eingesetzt werden und benötigen den Transportmanager für längenorientierte BE-Aufnehmer.

Als relevante Dimensionen werden die Länge in Transportrichtung und die Platzbreite abgebildet. Dabei können die Bausteine in vertikaler und horizontaler Ebene in beide Orientierungen ausgerichtet werden (siehe Abbildung 51). Es stehen also insgesamt vier Transportrichtungen zur Verfügung. Diese werden im Simulationssystem über die Richtungen 0, 90, 180 und 270 abgebildet.

Die Bauteilübergabe zwischen Plätzen wird geometriegerecht, unter Beachtung der Bauteildimensionen durchgeführt, d.h. die Bauteile befinden sich während der Übergabe auf beiden beteiligten Bausteinen (siehe Abbildung 51). Besitzen die beteiligten Plätze unterschiedliche Orientierungen, so werden die Bauteile, entgegen der eM-Plant-Standardmethodik, nicht gedreht. In diesem Fall wird die Ausdehnung in Transportrichtung mit der Ausdehnung quer dazu getauscht. Der benötigte Raum auf dem aufnehmenden Platz wird vor der Übergabe reserviert. Dies gilt auch, wenn der liefernde Bauteilnehmer Bauteile nur logisch verwaltet.

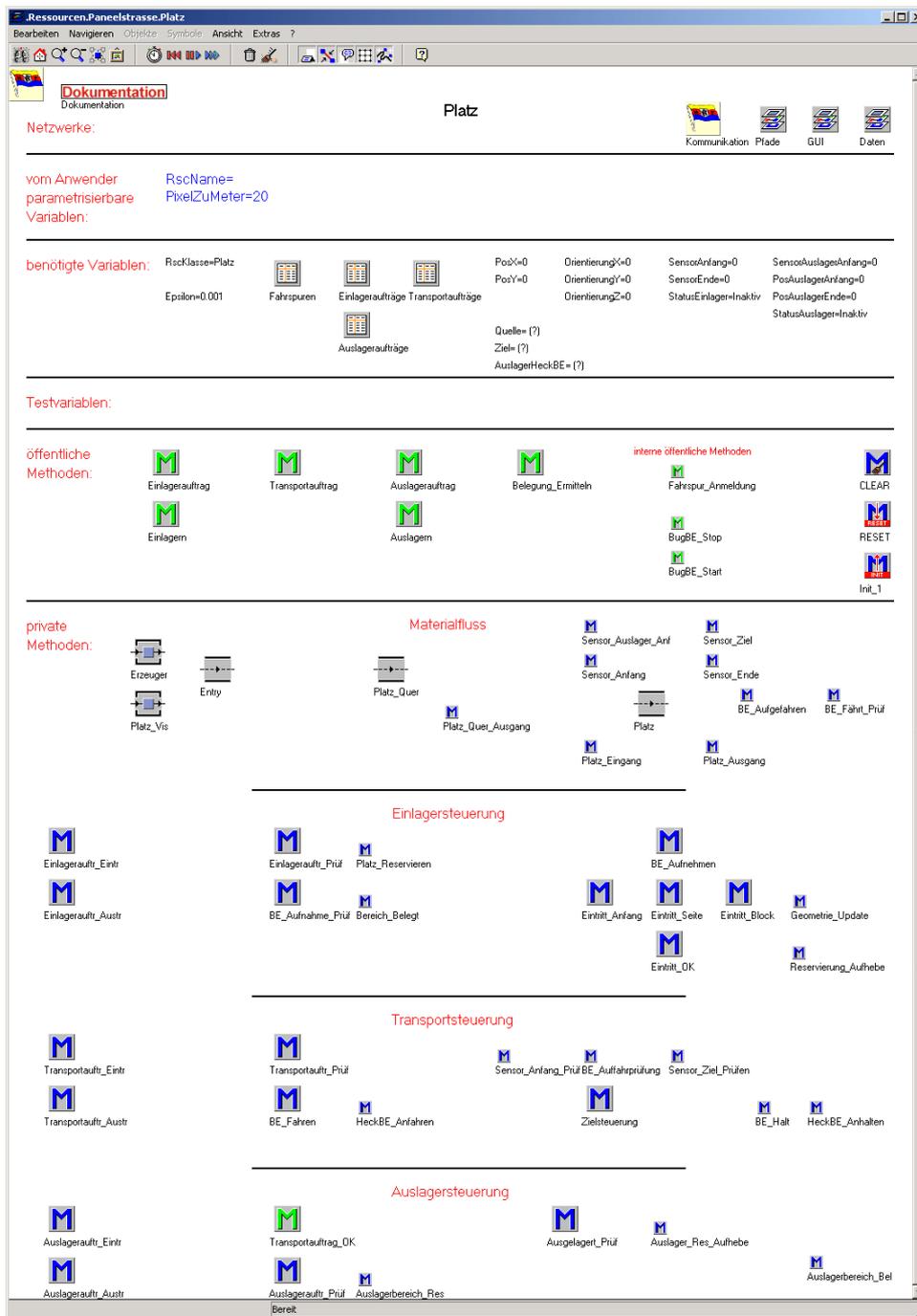


Abbildung 50 - Platz

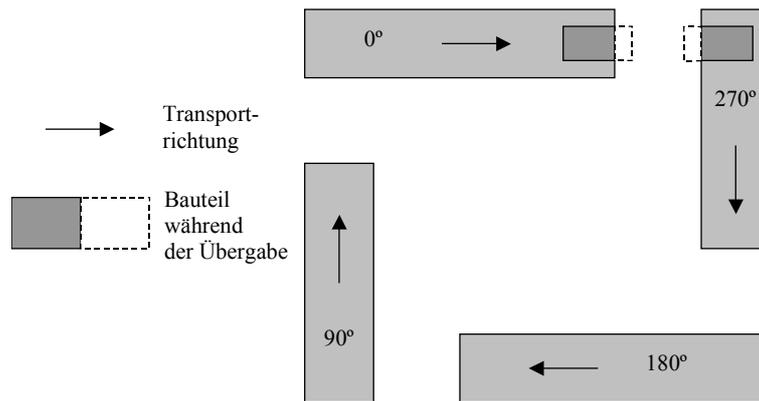


Abbildung 51 - Platz: Platzkopplung

Die Bauteile besitzen die in Bezug auf das Schiffskoordinatensystem absolut definierten Ausdehnungen Länge und Breite. Auf dem Platz wird entsprechend dem eM-Plant-Standardverhalten nur die Ausdehnung in Transportrichtung aktuell berechnet und verwaltet (siehe Abbildung 52). Bauteile können auf den Plätzen nicht neben- oder übereinander liegen.

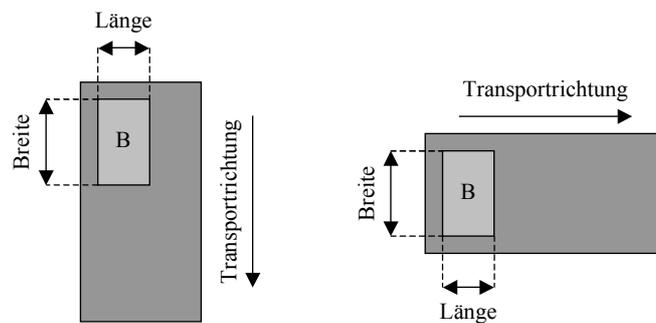


Abbildung 52 - Platz: Einsetzen des selben Bauteils auf Plätzen unterschiedlicher Orientierung

2.2.4.5.11 Fahrspur

Die Fahrspur bildet die Schnittstelle zwischen der beweglichen Maschine und den Plätzen. Jede bewegliche Maschine ist einer Fahrspur zugeordnet. Auf einer Fahrspur können sich mehrere Maschinen befinden. Die Fahrspur selbst ist einem Platz zugeordnet, ihre geometrische Lage wird aber absolut zum Fahrspurstandort angegeben. Die der Fahrspur zugeordneten beweglichen Maschinen können nur Bauteile bearbeiten die sich auf diesem Platz befinden.

Struktur

Die Maschinen auf einer Fahrspur könnten kollidieren (siehe Abbildung 53). Dies wird durch die implementierte Steuerung verhindert. Maschinen, die sich auf verschiedenen Fahrspuren befinden, können sich unabhängig voneinander bewegen.

Layoutübersicht

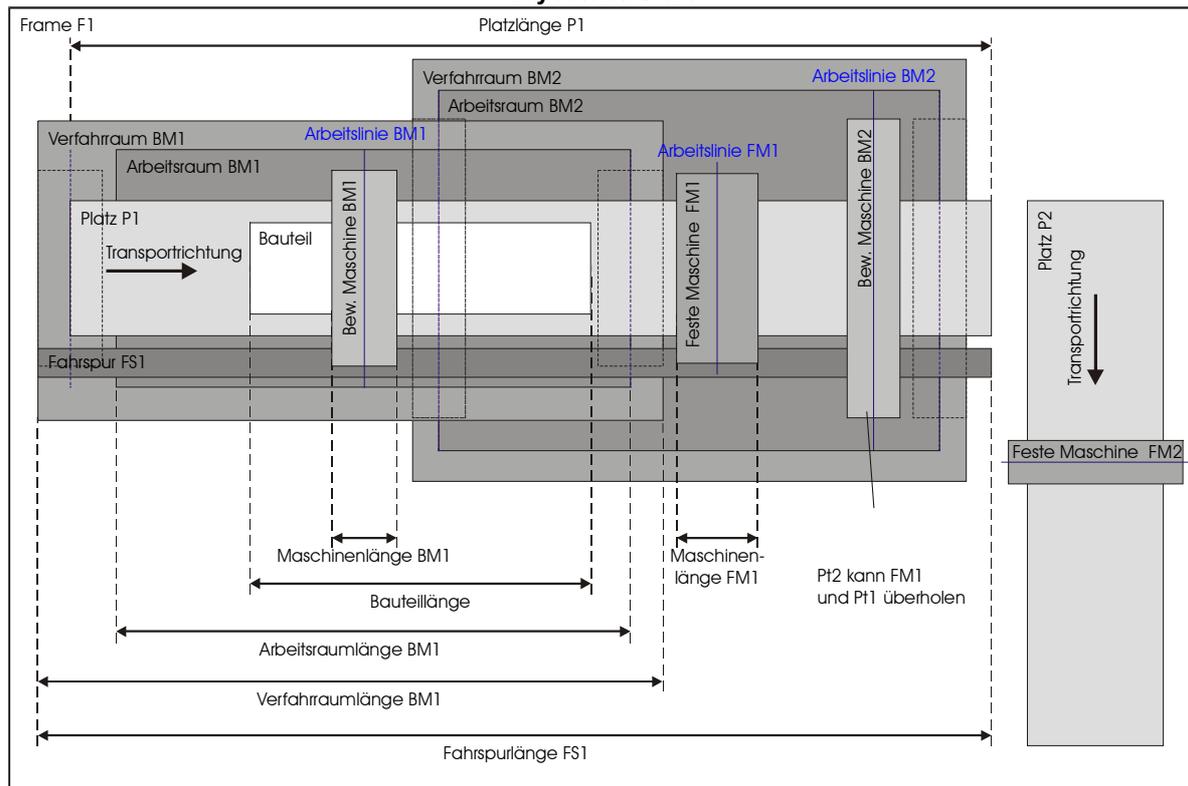


Abbildung 53 - Fahrspur: Fahrspurlayout

2.2.4.5.12 Maschine_Beweglich

Die in bewegliche Maschine ist die komplexeste im Toolset abgebildete Arbeitsressource und dient der Bearbeitung von Bauteilen. Sie stellt über ihren Ressourcenmanager Arbeitsleistung zur Verfügung. Die bewegliche Maschine ist im Gegensatz zu den anderen arbeitleistenden Ressourcen immer einer Fahrspur zugewiesen, welche wiederum einem Platz zugeordnet ist. Diese indirekte Zuordnung der beweglichen Maschine zu einem Platz wird durch den Ressourcenfinder berücksichtigt.

Nur die Geometrie in Transportrichtung wird als relevante Dimension abgebildet. Die Maschinen erhalten eine jeweils definierbare Länge in Transportrichtung mit einer mittig zentrierten Arbeitslinie. Arbeitsvorgänge können nur unter dieser Linie stattfinden. Die Maschinen werden an jeweils eine Fahrspur angegliedert und bekommen einen relativ zur Fahrspur angegebenen Arbeitsraum zugewiesen. Ihre Arbeitslinien können den Arbeitsraum nicht verlassen.

Die Arbeitsräume der Maschinen können sich überlappen. Um das Kollidieren von Maschinen zu verhindern, ist eine Kollisionsüberwachung implementiert. Diese berücksichtigt neben den zugewiesenen Maschinenarbeitsräumen auch deren Ausdehnungen, die zu vergrößerten Kollisionsräumen führen.

2.2.4.6 Öffentliche Methoden der Ressourcenklassen

Die öffentlichen Methoden der implementierten Ressourcenklassen sind zum Teil standardisiert und in mehreren Ressourcenklassen vertreten, zum Teil erfüllen sie sehr spezialisierte Funktionen und sind daher nur in jeweils einer Ressourcenklasse zu finden. Aus Tabelle 5 geht hervor, welche Öffentlichen Methoden in den einzelnen Ressourcenklassen eingesetzt sind.

	Quelle	Senke	Con- tainer	Werker	Werk- zeug	Masch' Stat.	Masch' Fest	Platz	Fahr- spur	Masch' Bew.
Auslagerauftrag	X		X				X	X		X
Auslagern	X		X				X	X	X	X
Einlagerauftrag		X	X				X	X		X
Einlagern		X	X				X	X	X	X
Belegung_Ermitteln			X				X	X		X
Pause				X	X	X	X			X
Entpausieren				X	X	X	X			X
Transportauftrag								X		
Fahrspur_Anmeldung								X		
BugBE_Stop								X		
BugBE_Start								X		
Maschine_Anmeldung									X	
Maschine_Verfahre									X	
Posit_Auftrag										X
Posit_Frage										X
Posit_Freigabe										X
Status_Ermitteln										X
Arbeitsraum_Info										X
M_Ziel_Erreicht										X
M_Umgebung_Info										X
Verschiebe_Auftrag										X
Verschieb_Auftr_OK										X

Tabelle 5 - Ressourcenklassen: Öffentliche Methoden der Ressourcenklassen

Es folgt eine Kurzbeschreibung der jeweiligen Funktion der öffentlichen Methoden.

- **Auslagerauftrag**

Über die Methode „Auslagerauftrag“ melden sich Elemente (z.B. der Transportmanager), um einen Auslagervorgang einzuleiten. Sobald Auslagerbereitschaft besteht, wird dem anfragenden Element "Auslagerauftrag_OK" gemeldet. Ggf. werden Auslageraufträge in einer FIFO-Liste gespeichert und der Reihenfolge nach ausgeführt.

Der Methode „Auslagerauftrag“ werden die Objektreferenzen des anfragenden Elements und des auszulagernden BEs als Parameter übergeben. Je nach Ressourcenklasse müssen ggf. zusätzlich die Austrittsart, die Austrittsseite und die Austrittsposition übermittelt werden.

- **Auslagern**

Nachdem dem anfragenden Element "Auslagerauftrag_OK" gemeldet wurde prüft das Element die Aufnahmebereitschaft des Zielobjekts. Sobald diese gegeben ist, ruft es die Methode „Auslagern“ auf, um das BE tatsächlich umzulagern. „Auslagern“ ruft daraufhin die Methode „Einlagern“ der Zielressource auf. Der Methode „Auslagern“ werden die Objektreferenzen der Zielressource und des auszulagernden BEs als Parameter übergeben.

Nur für Ressourcenklasse Fahrspur: Über die interne öffentliche Methode „Auslagern“ können Bauteile vom Maschinen-BE auf eine Zielressource umgelagert werden.

- **Einlagerauftrag**

Über die Methode „Einlagerauftrag“ melden sich Elemente (z.B. der Transportmanager), um einen Einlagervorgang einzuleiten. Sobald Einlagerbereitschaft besteht, wird dem anfragenden Element "Einlagerauftrag_OK" gemeldet. Ggf. werden mehrere wartende Einlageraufträge in einer FIFO-Liste gespeichert und der Reihenfolge nach ausgeführt.

Der Methode „Einlagerauftrag“ werden die Objektreferenzen des anfragenden Elements und des einzulagernden BEs als Parameter übergeben. Je nach Ressourcenklasse müssen ggf. zusätzlich die Eintrittsart, die Eintrittsseite und die Eintrittsposition übermittelt werden.

- **Einlagern**

Sobald der angefragte Baustein dem Anfrager "Einlagerauftrag_OK" gemeldet hat, wird über diese Methode der konkrete Einlagerungsvorgang durchgeführt. Der Methode „Einlagern“ werden die Objektreferenzen der Quellressource und des einzulagernden BEs als Parameter übergeben.

Nur für Ressourcenklasse Fahrspur: Über die interne öffentliche Methode „Einlagern“ werden Bauteile auf das Maschinen-BE umgelagert.

- **Belegung_Ermitteln**

Über die Methode „Belegung_Ermitteln“ können Informationen über BEs-aufnehmende Ressourcen und ihre Belegung abgerufen werden. Der Methode werden keine Parameter übergeben. Je nach angefragter Ressourcenklasse enthält das zurückgegebene Datenpaket die Anzahl der BEs auf der Ressource, die von BEs belegte Länge etc.

- **Pause**

Über die Methode „Pause“ können externe Elemente (z.B. der Pausen-Baustein) qualifikations anbietende Ressourcen pausieren. Sie leitet die Anforderung an den Ressourcenmanager über die Meldung „Verwendung unterbrechen“ weiter.

- **Entpausieren**

Über die Methode „Entpausieren“ können externe Elemente (z.B. der Pausen-Baustein) qualifikations anbietende Ressourcen entpausieren. Sie leitet die Anforderung an den Ressourcenmanager über die Meldung „Verwendung wiederaufnehmen“ weiter.

- **Transportauftrag (nur Ressourcenklasse Platz)**

Die Förderung von Bauteilen wird nach Aufruf der Methode „Transportauftrag“ (unter Angabe der minimalen und maximalen Zielposition) eigenständig vom Platz durchgeführt. Es können mehrere Transportaufträge gleichzeitig ausgeführt werden. Die Bauteile

werden soweit wie möglich in Förderrichtung bewegt. Zwei Bedingungen können den Transport beenden:

- Die maximale Zielposition wurde erreicht
- Das Bauteil ist auf ein anderes Bauteil aufgefahren und hat die minimale Zielposition überschritten

Sobald eine der Bedingungen zutrifft, wird die Methode „Transportauftrag_OK“ des auftraggebenden Elements aufgerufen.

Der Methode „Transportauftrag“ werden die Objektreferenzen des auftraggebenden Elements und des zu transportierenden BEs als Parameter übergeben. Zusätzlich müssen die minimal und die maximal zu erreichende Position übergeben werden.

- **Fahrspur_Anmeldung (nur Ressourcenklasse Platz)**

Über die interne öffentliche Methode „Fahrspur_Anmeldung“ melden sich die dem Platz zugewiesenen Fahrspuren in der Init-Phase an.

- **BugBE_Stop (nur Ressourcenklasse Platz)**

Über die interne öffentliche Methode „BugBE_Stop“ melden nachfolgende Bauteilaufnehmer den Stop eines BEs, welches sich mit seinem Heck noch auf dem Platz befindet.

- **BugBE_Start (nur Ressourcenklasse Platz)**

Über die interne öffentliche Methode „BugBE_Start“ melden nachfolgende Bauteilaufnehmer den Start eines BEs, welches sich mit seinem Heck noch auf dem Platz befindet.

- **Maschine_Anmeldung (nur Ressourcenklasse Fahrspur)**

In der Init-Phase melden sich die zugeordneten beweglichen Maschinen über die interne öffentliche Methode „Maschine_Anmeldung“ bei der Fahrspur an. Die Fahrspur enthält ein eM-Plant-Weg-Element „Platz“, das Maschinen-BEs als Repräsentanten der beweglichen Maschinen aufnimmt.

- **Maschine_Verfahre (nur Ressourcenklasse Fahrspur)**

Die Fahrspur nimmt Verfahraufträge über die interne öffentliche Methode „Maschine_Verfahre“ entgegen und meldet deren Durchführung an die aufrufende bewegliche Maschine (Methode „M_Ziel_Erreicht“).

- **Posit_Auftrag (nur Ressourcenklasse Maschine_Beweglich)**

Die bewegliche Maschine nimmt Verfahrbefehle über die Methode „Posit_Auftrag“ entgegen. Sie prüft die Durchführbarkeit des Auftrags:

- Der Verfahrraum ist frei

Der Verfahrbefehl ist direkt durchführbar. Die bewegliche Maschine gibt der Fahrspur den Befehl das Maschinen-BE in Bewegung zu setzen. Ist die Position erreicht, so meldet sich die Fahrspur bei der beweglichen Maschine. Sie leitet daraufhin die Information an das ursprünglich befehlsgebende Element weiter (Methode „Posit_Auftrag_OK“).

- Der Verfahrraum ist blockiert

Kann der Verfahrbefehl nicht ausgeführt werden, da sich eine Nachbarmaschine im Weg befindet, so wird dieser Maschine ein Verschiebeauftrag gegeben. Die Nachbarmaschine meldet sich, sobald sie den Verschiebeauftrag durchgeführt hat. Ein Verschiebeauftrag an die linke (rechte) Nachbarmaschine muss auch gegeben werden, wenn ein Verschiebeauftrag von der rechten (linken) Nachbarmaschine nicht direkt durchgeführt werden kann.

Der Methode wird die Objektreferenz des auftraggebenden Elements, eine Auftragsnummer, die Zielposition und die Auftragspriorität übergeben.

- **Posit_Frage (nur Ressourcenklasse Maschine_Beweglich)**

Über die Methode kann festgestellt werden ob ein Positionierungsbefehl durchgeführt wurde. Ihr wird dazu die Objektreferenz des auftraggebenden Elements und die Auftragsnummer übergeben.

- **Posit_Freigabe (nur Ressourcenklasse Maschine_Beweglich)**

Die Methode dient der Freigabe der durch einen Positionierungsauftrag fixierten Resource. Sind weitere Aufträge vorhanden, so wird der mit der höchsten Priorität und der längsten Wartezeit eingelastet. Der Methode wird die Objektreferenz des auftraggebenden Elements und die Auftragsnummer übergeben.

- **Status_Ermitteln (nur Ressourcenklasse Maschine_Beweglich)**

Über die Methode „Status_Ermitteln“ können Informationen über Status und Konfiguration der beweglichen Maschine abgerufen werden. Der Methode werden keine Parameter übergeben. Folgende Einträge enthält das zurückgegebene Datenpaket:

Zeilenindex	Spaltenindex	Inhalt
Maschinenlänge	Real	Länge der Maschine in mm
Geschwindigkeit	Real	Geschwindigkeit der Maschine in mm / s
ArbeitsraumAnfangR	Real	Arbeitsraumanfang unter Berücksichtigung der Nachbarmaschinenarbeitsräume in mm
ArbeitsraumEndeR	Real	Arbeitsraumende unter Berücksichtigung der Nachbarmaschinenarbeitsräume in mm
Fahrspur	Object	Zugewiesene Fahrspur
Mlinks	Object	Linke Nachbarmaschine
Mrechts	Object	Rechte Nachbarmaschine
Status	String	Status der Maschine: - Stop (Kein Auftrag) - Job (Positionierungsauftrag wird durchgeführt) - MLinks (Verschiebeauftrag der linken Nachbarmaschine wird durchgeführt) - MRechts (Verschiebeauftrag der rechten Nachbarmaschine wird durchgeführt)
Position	Real	Position der Maschinenmitte
PauseAktiv	Boolean	Flag das Pausenstatus wiedergibt

Tabelle 6 - Meldungen Maschine_Beweglich: Rückgabe Status_Ermitteln

- **Arbeitsraum_Info (nur Ressourcenklasse Maschine_Beweglich)**

In der Init-Phase liefert die zugeordnete Fahrspur Arbeitsrauminformationen über die interne öffentliche Methode „Arbeitsraum_Info“ an die bewegliche Maschine.

- **M_Ziel_Erreicht (nur Ressourcenklasse Maschine_Beweglich)**
Die Fahrspur meldet erfüllte Verfahrtaufträge über die interne öffentliche Methode „M_Ziel_Erreicht“.
- **M_Umgebung_Info (nur Ressourcenklasse Maschine_Beweglich)**
Die Nachbarmaschinen geben Information über ihre neuen Positionen über die interne öffentliche Methode „M_Umgebung_Info“ an die bewegliche Maschine.
- **Verschiebe_Auftrag (nur Ressourcenklasse Maschine_Beweglich)**
Verschiebeaufträge von Nachbarmaschinen werden über die interne öffentliche Methode „Verschiebe_Auftrag“ übermittelt.
- **Verschieb_Auftr_OK (nur Ressourcenklasse Maschine_Beweglich)**
Die Erfüllung von Verschiebeaufträgen melden die Nachbarmaschinen über die interne öffentliche Methode „Verschiebe_Auftrag_OK“.

2.2.4.7 Transportmanager



Der Transportmanager dient sowohl als BE- oder Materialflussschnittstelle eines Netzwerks zur weiteren Simulationsumgebung, als auch der Steuerung des Transports zwischen den Ressourcen eines Netzwerks. In jedem Fertigungsnetzwerk muss ein Transportmanager instanziiert werden.

Es handelt sich bei dem hier erläuterten Element um den Transportmanager zur Verwaltung von längenorientierten BE-Aufnehmern, wie z.B. dem Platz.

Die Schnittstellen zu den BE-Aufnehmenden Ressourcen des verwalteten Netzwerks sind dementsprechend angepasst. Die Schnittstellen zu den externen Managern, dem Leitreechner und auch dem internen Jobmanager entsprechen jedoch, um die Kompatibilität zu wahren, dem Standard.

Öffentliche Methoden

- **Ressource_Anmeldung**
In der Init-Phase melden sich die BE-Aufnehmenden Ressourcen beim Transportmanager über die Methode „Ressource_Anmeldung“ an. Sie übermitteln ihren Namen, ihre Klasse und ihren Objektpfad. Die Methode leitet die Anmeldung an den zentralen Leitreechner weiter.
- **Ressource_Abfrage**
Über die Methode können andere Elemente unter Angabe des Ressourcennamen und der Ressourcenklasse den Objektpfad einer Ressource ermitteln.
- **BE_Erzeugt**
Ressourcen, auf denen ein BE erzeugt wurde, melden dies dem Transportmanager über die Methode „BE_Erzeugt“. Als einziger Parameter wird die Objektreferenz des BEs übergeben. Die Methode gibt die Meldung an den Jobmanager weiter, der den Leitreechner informiert.
- **Transport_Auftrag**

Der Auftrag zum Transport eines BEs wird dem zuständigen Transportmanager über die Methode „Transport_Auftrag“ gegeben. Dem als Referenz übergebenen BE muss die Transportwegtabelle und das ausgewählte Paket als Attribut angehängt sein.

- **Auslagerauftrag_OK**

Wurde einer Ressource ein Auslagerauftrag für ein BE erteilt, so ruft diese bei Auslagerbereitschaft die Methode „Auslagerauftrag_OK“ auf. Als einziger Parameter wird die Objektreferenz des BEs übergeben.

- **Einlagerauftrag**

„Einlagerauftrag“ wird von der Methode „Auslagerauftrag_OK“ oder von Transportmanagern anderer Netzwerke aufgerufen. Externe Manager wenden sich mit dem Einlagerauftrag nicht direkt an die Zielressourcen, sondern sprechen zunächst den zuständigen Transportmanager an. Der hier beschriebene Manager ermittelt die spezifischen Einlagerparameter für längenorientierte BE_Aufnehmer (z.B. für den Platz) und gibt die Anfrage dann an die Ressource weiter. Als Parameter werden die Objektreferenz des anfragenden Elements und die des BEs übergeben.

- **Einlagerauftrag_OK**

Wurde einer Ressource oder einem anderen Transportmanager ein Einlagerauftrag für ein BE erteilt, so ruft diese bzw. dieser bei Einlagerbereitschaft die Methode „Einlagerauftrag_OK“ auf. Als einziger Parameter wird die Objektreferenz des BEs übergeben.

- **Eingelagert**

Die Einlagerung eines BEs wird dem zuständigen Transportmanager von der aufnehmenden Ressource über die Methode „Eingelagert“ angezeigt. Als einziger Parameter wird die Objektreferenz des BEs übergeben.

- **Posit_Auftrag**

Der Jobmanager sendet vor jeder Nutzung einer Ressource zunächst einen Positionierungsauftrag für die Ressource an den lokalen Transportmanager. Der hier beschriebene Transportmanager ist für Ressourcen zuständig, die sich vor der Prozessausführung noch positionieren müssen. Die Meldung wird daher an die jeweilige Ressource weitergegeben. Das der Methode übergebene Datenpaket enthält die Objektreferenzen des anzusteuernenden BEs und der zu positionierenden Ressource (siehe Meldung „Posit Auftrag“).

- **Posit_Auftrag_OK**

Über die Methode „Posit_Auftrag_OK“ melden sich die Ressourcen sobald sie sich positioniert haben. Der Methode wird die Objektreferenz der Ressource und die von „Posit_Auftrag“ vergebene Auftragsnummer übermittelt. Sie meldet die erfolgte Positionierung an den lokalen Jobmanager weiter.

- **Posit_Freigabe**

Der Jobmanager sendet nach jeder Freigabe einer Ressource eine Positionsfreigabemeldung an den lokalen Transportmanager. Die Meldung wird an die jeweilige Ressource weitergegeben. Das der Methode übergebene Datenpaket enthält die Objektreferenzen des angesteuerten BEs und der positionierten Ressource (siehe Meldung „Posit Freigabe“).

2.2.4.8 BEs

Die hier beschriebenen BEs dienen der Abbildung der Bauteile, beweglichen Maschinen und zu allgemeinen Visualisierungszwecken. Sie sind im Ordner Hilfsklassen hinterlegt.

- **BE**

Bauteile können über das BE (abgeleitet von der eM-Plant Fahrzeug-Klasse) abgebildet werden. Es sollte bei Simulationsmodellen zum Einsatz kommen, die Ressourcen der Platz-Klasse beinhalten, da es entsprechende Visualisierungssymbole zur Verfügung stellt.

- **Balken**

Der vom Fördergut abgeleitete Balken wird nur für Visualisierungszwecke eingesetzt (z.B. Anzeige der Arbeitsraums der beweglichen Maschinen oder verschiedene Statusindikatoren).

- **Maschine**

Das vom Fahrzeug abgeleitete BE „Maschine“ wird auf der Fahrspur eingesetzt. Zu jedem Objekt „Maschine_Beweglich“ wird auf der zugeordneten Fahrspur ein BE „Maschine“ erzeugt.

2.2.4.9 Erstellen von Simulationsmodellen

Im folgenden werden die Grundzüge beim Aufbau von Simulationsmodellen mit dem Toolset aufgezeigt.

- **Aufbau der Grundstruktur**

Das Modell wird in einem beliebigen eM-Plant Netzwerk aufgebaut. In dieses werden folgende Komponenten eingesetzt:

- obligatorisch: Leitreechner, Ereignisverwalter, Ressourcennetzwerk (s.u.)
- optional: Pausenbaustein, Verfügbarkeitsmanager

Die Komponenten kommunizieren mit diversen Bausteinen, die in der Klassenbibliothek vorhanden und den Komponenten bekannt sein müssen:

- obligatorisch: Eventschnittstelle, Arbeitsplanmanager, Belastungsprüfung, Prozesszeitmanager, Ressourcenfinder, Transportplaner, Datenbaustein
- optional: Data-Baustein, Koordinaten-Komponente, SIMPLEMethoden-Komponente, weitere Auswertungsbausteine, weitere Visualisierungsbausteine

- **Aufbau der Ressourcennetzwerkstruktur**

Dem Modell kann eine hierarchische Struktur (z.B. organisatorisch oder räumlich zusammenhängende Abteilungen) gegeben werden, indem in das Ressourcennetzwerk wiederum Netzwerke beliebig tief verschachtelt eingesetzt werden.

In alle Ressourcennetze werden anschließend jeweils ein Jobmanager und ein Transportmanager instanziiert.

- **Einsetzen der Ressourcen**

Alle im Simulationsmodell abzubildenden Ressourcen werden anschließend im Ressourcennetzwerk instanziiert. Folgende Elemente stehen zur Verfügung:

- BE-Aufnehmer (Container, Platz)
- Arbeitsanbieter (Werker, Werkzeug, Maschine_Stationär, Maschine_Beweglich)
- Hybridressourcen (Maschine_Fest)
- Spezielle Ressourcen (Quelle, Senke, Fahrspur)

Jedes Modell muss mindestens eine Quelle und eine Senke enthalten. In jedem Netzwerk muss mindestens ein BE-Aufnehmer und eine arbeitleistende Ressource (oder eine Hybridressource) vorhanden sein. Eine bewegliche Maschine kann nur im Zusammenhang mit einer Fahrspur und einem Platz eingesetzt werden.

- **Parametrierung der Modellelemente**

- Alle Modellelemente: Alle Modellelemente, die ein Netzwerk „Pfade“ besitzen, müssen parametrierbar sein: In dem Netzwerk finden sich Variablen, die auf die von dem Element benötigten Komponenten zeigen.
- Alle Ressourcen: Die eingesetzten Ressourcen müssen benannt werden. Dazu dient in jeder Ressource die Variable „RscName“. Der Name und die Bezeichnung der Ressourcenklasse ergeben den vollständigen Schlüssel.
- Quellen: Die Lieferlisten der Quellen sind zu füllen. Als Attribute sind die Topologie („ID_Topologie“) und die Sicht („Sicht“) des BEs in der Attribute-Untertabelle anzugeben.

- **Erstellen der Datenbasis**

Nach Fertigstellung der Modellstruktur wird die Datenbasis der Simulation definiert. In den Tabellen des Datenbausteins müssen die für die Simulation notwendigen Informationen (Produkt-, Arbeitsplan- und Ressourcenmodell, 2.2.3) eingetragen bzw. aus einer Datenbank eingelesen werden.

- **Modellbeispiel**

Ein auf dem Toolset beruhendes, einfaches Simulationsmodell ist in Abbildung 54 dargestellt. Es zeigt exemplarisch den grundlegenden Aufbau von Toolsetmodellen.

- Im Hauptordner befindet sich neben Pausenbaustein, Verfügbarkeitsmanager, Leit-rechner und Ereignisverwalter auch das Ressourcennetzwerk „Panneelstraße“.
- Job- und Transportmanager des Fertigungsnetzwerks übernehmen die Steuerungsaufgaben bezüglich Ressourcenverwaltung, Auftragsbearbeitung und BE-Transport. Die Ressourcen des Netzwerks sind:
 - ♦ zwei Werker
 - ♦ zwei bewegliche Maschinen, die sich auf einer gemeinsamen Fahrspur befinden
 - ♦ ein Platz, dem die Fahrspur zugeordnet ist

Weiterhin befindet sich ein Subfertigungsnetzwerk „Lager“ in dem Fertigungsnetzwerk. Das Subfertigungsnetzwerk enthält wiederum eigenständige Job- und Transportmanager. Die Ressourcen des Netzwerks sind:

- ◆ zwei Werker
- ◆ zwei Container

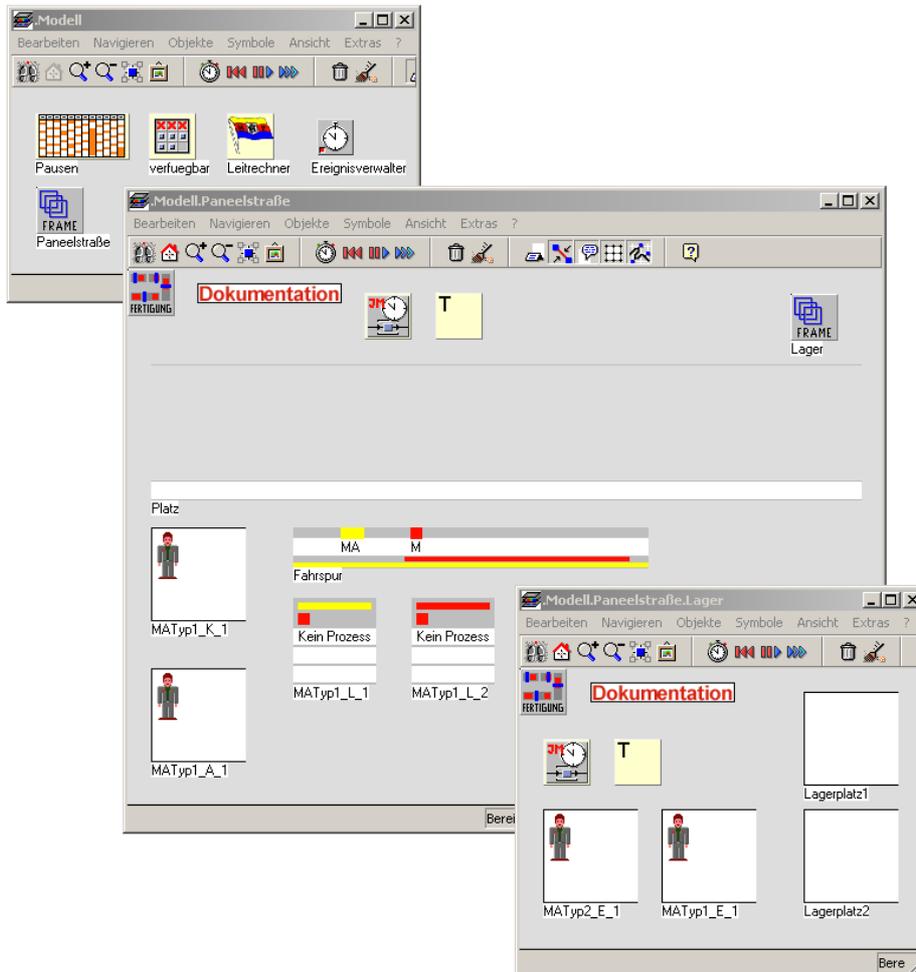


Abbildung 54 - Modelle erstellen: Beispiel

2.2.4.10 Erstellen neuer Toolsetklassen

Das Toolset lässt sich um neue Ressourcenklassen erweitern. Die Ressourcen setzen sich, je nach Funktionstyp, aus verschiedenen Standardelementen zusammen. Die eingesetzten Ressourcenelemente definieren dabei den Typ der Ressource. Es können sowohl BE-Aufnehmer, Arbeitsanbieter, Hybridelemente als auch benutzerdefinierte Ressourcen entwickelt werden. In der Klassenbibliothek stehen dafür diese Vorlagen zur Verfügung:

- *Rsc_Vorlage*
Allgemeine Ressourcenvorlage
- *Rsc_Vorlage_ArbAnb*
Arbeitsleistende Ressource

- *Rsc_Vorlage_BEAufn*
BE aufnehmende Ressource
- *Rsc_Vorlage_Hybrid*
Arbeitsleistende und BE aufnehmende Ressource

Die Vorlagen enthalten entsprechend der jeweiligen Funktion erforderliche und optionale Elemente. Die Regeln zur Strukturierung, zur Implementierung spezieller Funktionalitäten, zur Verankerung einer neuen Ressourcenklasse im Datenbaustein etc. wurden dokumentiert, werden aber an dieser Stelle nicht weiter aufgeführt.

2.2.5 Raumbaustein

Flächen (Bauplätze) stellen oftmals eine Engpassressource für ein Schiffbauprojekt dar. Eine genaue Planung der Flächenbelegung ist somit eine sehr wichtige Planungsaufgabe. Im Projekt wurde für diese Planungsaufgabe eine simulationsbasierte Planungslösung entwickelt. Für eine Kompaktwerft wie die Meyer Werft ist die Simulation der Flächenbelegung (Raumbelegung) ist eine wichtige Anwendung.

Für diesen komplexen Themenbereich wurde im Projekt ein Baustein entwickelt, mit dem sowohl einzelne Bereiche (Flächen) einer Werft, als auch die gesamte Werft simuliert werden kann. Die Simulation muss auf der Basis der Geometrie der einzelnen Bauteile (Sektionen, Blöcke) und der realen Abmessungen der Flächen (Bauplätze) erfolgen. Eine Simulation nur auf Basis von Bauplätzen ist für eine Kompaktwerft zu ungenau. Die Entwicklung eines vierdimensionalen Schachtelalgorithmus (Geometrie und zeitlicher Verlauf) war eine erhebliche Herausforderung für das Simulationsprojekt.

Mit der Entwicklung des Raumbausteins auf Simulationsebene wurde zu Beginn des Projektes begonnen. Während des gesamten Projektes wurde der Baustein gepflegt und gewartet. Die stetige Weiterentwicklung war durch den modularen Aufbau der Komponente möglich. Der Raumbaustein teilt sich in die Komponenten logische Elemente und Funktionen, Grundbelegung, Dialog und Visualisierung auf:

2.2.5.1 Logische Elemente und Funktionen

In den logischen Elementen wurden alle Funktionen, Algorithmen, Eingangsdaten etc. abgebildet, die für die Verwaltung der Bauteile in den Räumen erforderlich sind. Folgende Funktionen, Algorithmen und Daten wurden im Baustein hinterlegt:

- Integration des Datenmodells
Für die Anbindung des Raumbausteins an das Datenmodell werden die Datenbausteine verwendet.
- Ein-, Aus-, und Umlagern von Bauteilen
Bauteile müssen in die Hallen ein- und ausgelagert werden. Während der Bearbeitung der Bauteile kann eine Umlagerung des Bauteils auf einen anderen Bauplatz erforderlich sein.
- Montage von Bauteilen in Räumen
Bauteile werden auf den Flächen montiert. Sektionen werden auf Bauplätzen zu Blöcken montiert. Blöcke werden im Baudock zum Schiffskörper montiert.
- Bearbeitung von Bauteilen in Räumen ohne Montage
Bauteile werden ohne Montage auf Bauplätzen bearbeitet (z.B. Ausrüstung)
- Definition von gesperrten Räumen

Bauflächen in einer Halle unterliegen einigen Restriktionen. So können Bauflächen gesperrt werden (z.B. Ablage- und Logistikflächen). Des weitern ist eine zeitliche Änderungen der Sperrungen möglich

- Definition der Abmessungen von Räumen
Abmessungen von Hallen (Geometrie der Baufläche) müssen modelliert werden (zeitliche Änderungen der Abmessungen sind möglich).
- Drehen und Stapeln von Bauteilen
Bauteile werden vor der Einlagerung auf dem Bauplatz um 90° gedreht oder zwei Bauteile werden auf einem Bauplatz gestapelt
- Überschreiben von gesperrten Flächen
Gesperrte Flächen dürfen in einigen Situationen überschrieben werden.
- Speicherung des Lifecycle für jeden Raum
Der simulierte Belegungsplan muss im Datenmodell gespeichert werden.
- Schnittstellen
Im Raumbaustein wurden neben den klassischen Möglichkeiten zur Hinterlegung von Ein- und Ausgangssteuerungen, weitere Eingriffsmöglichkeiten – Interfaces – zur Einbindung von Steuerungen geschaffen. Durch diese Interfaces kann das Verhalten im Raumbaustein individuell gesteuert werden.
- Bausteinspezifische Statistiken
Für Auswertungen muss eine raumspezifische Statistik erzeugt werden.

Besonders Funktionen und Algorithmen der logischen Ebene wurden modular aufgebaut, so dass Änderungen einzelner Funktionen und Algorithmen jederzeit möglich sind. Abbildung 55 zeigt die eine Übersicht über den Raumbaustein.

2.2.5.2 Dialog

Die Parametrierung und die Steuerung des Bausteins auf Programmiererebene soll durch einen Dialog (Standard für alle Simulationsbausteine von eM-Plant) erfolgen. Mit dem Dialog wird das Verhalten des Raums im Simulationsmodell durch den Programmierer festgelegt. Dadurch sind keine Eingriffe in das Objekt bzw. im Programmcode des Raumbausteins erforderlich, eine individuelle Anpassung des Bausteins auf das jeweilige Modell jedoch möglich. Im Dialog werden z.B. die Abmessungen der Räume, die gesperrten Flächen, die Statistik etc. parametriert bzw. aktiviert. Steuerungen (Methodenaufrufe) können im Dialog, wie in eM-Plant Standardelementen hinterlegt werden. Die angestrebte Anwenderfreundlichkeit konnte so erreicht werden.

2.2.5.3 Visualisierung

Eine 3D-Visualisierung der Raumbelugung ist von elementarer Bedeutung. Schon in der Testphase zeigte sich, dass nur durch eine Visualisierung das Verhalten des Bausteins nachvollzogen und getestet werden kann. Abbildung 55 zeigt die 3 D-Darstellung der Zulage der Meyer Werft.

Die Visualisierung der Bauteile erfolgt auf der Basis des entwickelten Produktmodells. Die Visualisierung kann dabei in drei im Dialog wählbaren Stufen erfolgen. In der ersten, sehr groben Stufe erfolgt die Visualisierung der Bauteile auf Basis der Boundingbox (Umhüllende) der Bauteile. In der zweiten Stufe erfolgt die Visualisierung auf der Basis der Acht-Punkte-Geometrie. Für diese Visualisierung wurde im 3D-Viewer von eM-Plant der Befehl „Setindexfaceset“ integriert. Mit diesem Befehl wird eine Visualisierung der Bauteile auf Basis des Geometriemodells ermöglicht. In der dritten, sehr feinen Stufe ist eine Visualisierung der

Bauteile durch eine graphische Repräsentanz mit Hilfe von einzelnen VRML Dateien vorgehen. Diese Dateien müssen von CAD System zur Verfügung gestellt werden. In der Praxis wird meistens die mittlere Stufe für die Visualisierung verwendet.

2.2.5.4 Lifecycle und Grundbelegung

Neben der Visualisierung der Bauteile in den Hallen ist die Statistik und der "Life Cycle" der Hallen eine Möglichkeit zur Auswertung von Simulationsexperimenten. Im "Life Cycle" werden alle Aktionen (Einlagerung, Auslagerung und Montage von Bauteilen, Lage von gesperrten Räumen etc.), die einen Raum in einem Simulationsmodell betreffen, dokumentiert. Für den Ergebnistransfer wurden eine "Lifecycle" Datenbank, die Schnittstelle und Auswertungstools konzipiert und umgesetzt. Dadurch ist eine nachträgliche Auswertung und Visualisierung der Experimente (Simulationsvarianten) möglich.

Für die Ablaufplanung ist eine Vorbelegung von Flächen (verplante Flächen) erforderlich. Im Gegensatz zur klassischen Simulation kann bei der Flächenbelegung nicht mit Modellen gearbeitet werden, bei denen eine Einschwingphase des Modells erforderlich ist. Im Raumbaustein wurde ein Algorithmus entwickelt, der auf der Basis einer Grundbelegungsvariante vorbelegte Flächen reserviert und sie so für Bauteile, die neu in die Halle eingelagert werden sollen, sperrt. Der Ablauf wurde als Modul in den Raumbaustein integriert.

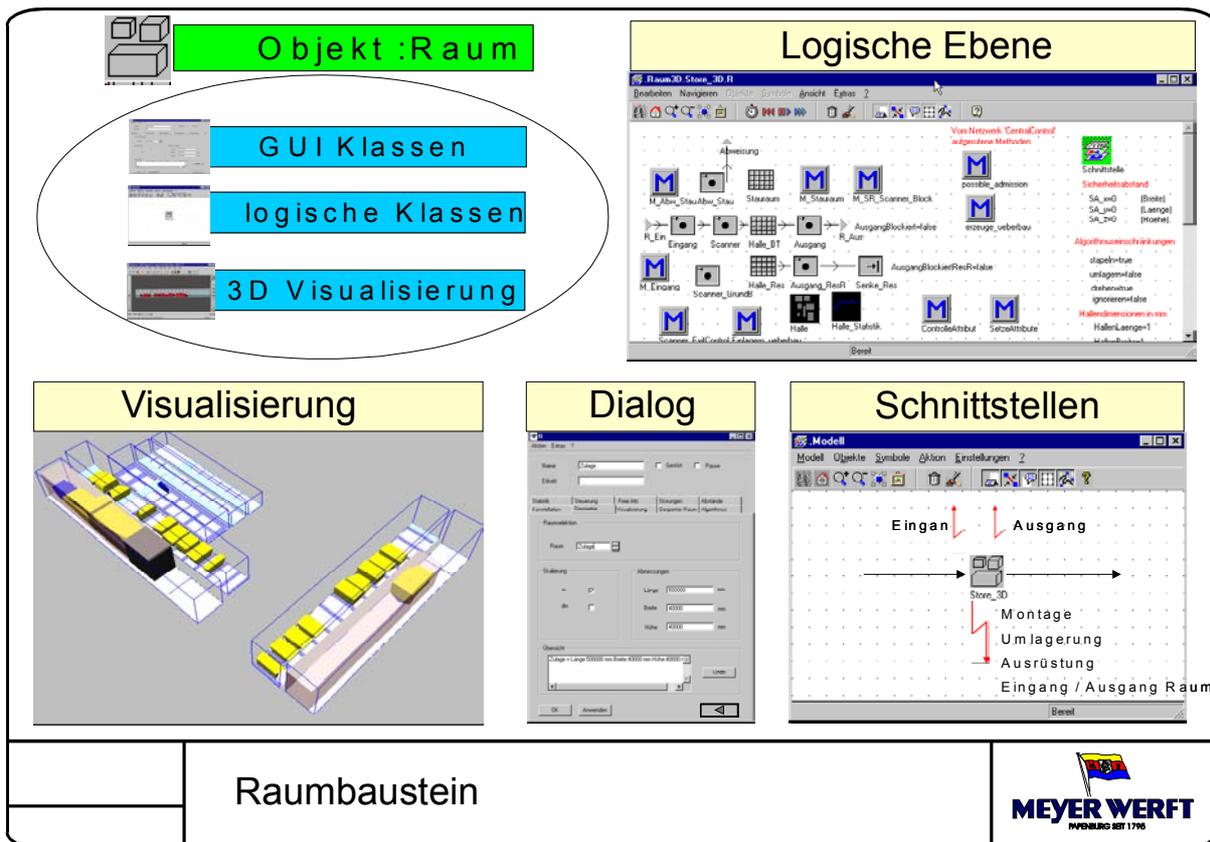


Abbildung 55 - Raumbaustein: Übersicht der Komponenten

2.2.6 Übergeordnete Steuerungen

Als übergeordnete Bausteine werden Komponenten aufgefasst, die in ein Simulationsmodell eingesetzt werden und auf den Simulationsablauf einwirken, ohne dass umgekehrt Simulationsereignisse Rückwirkungen auf das Verhalten dieser Steuerungen haben.

2.2.6.1 Pausenbaustein



Im Pausenbaustein werden planmäßige Arbeitszeitunterbrechungen wie Pausen, Schichten, Urlaubs- und Feiertage simulationsseitig abgebildet. Der Baustein bietet die Möglichkeit, entsprechende Schichtkalender zu modellieren. Er wird dazu eingesetzt, Materialflussbausteine im Simulationslauf - auch bei Rückwärtssimulation - gemäß Schichtkalender zu pausieren. Die Bearbeitung von Bauteilen durch diese Ressourcen in diesem Zustand ist dann unterbrochen bzw. kann nicht begonnen werden.

Außerdem bietet der Pausenbaustein öffentliche Methoden, die z.B. für die Terminierung unter Berücksichtigung von Schichtkalendern genutzt werden können.

Modellieren von Schichtkalendern

Damit der Pausenbaustein zur Steuerung von Ressourcen im Simulationsablauf eingesetzt werden kann, ist das Netzwerk „Pausen“ in das unterste Netzwerk der Modellhierarchie zu instanzieren. Mit einem Doppelklick auf das Bausteinsymbol wird der Hauptdialog geöffnet (Abbildung 56-A). Er bietet Funktionen zum Konfigurieren von Schichtmodellen, zu deren Einsatz in der Simulation und Zugriff auf eine Datenbank, in der Schichtkalender gepflegt werden. Die Funktionen sind im einzelnen:

Konfigurieren elementarer Logiken (Abbildung 56-B)

Es werden drei Typen elementarer Logiken unterschieden:

- Pausenlogiken: Hier werden Muster für den periodisch sich wiederholenden Ablauf von Arbeitstagen hinterlegt (Pausen, Schichtbeginn und -ende etc.). Zu jeder Logik wird eine Wertetabelle mit Zeitintervallen (Uhrzeit von - bis) und dem jeweiligen Pausiertheitszustand angegeben.
- Wochenlogiken: Hier werden Muster für den periodisch sich wiederholenden Ablauf von Arbeitswochen hinterlegt (5-, 6-Tage-Woche etc.). Zu jeder Logik wird eine Wertetabelle mit Zeitintervallen (Tag von - bis) und der am Tag jeweils anzuwendenden Pausenlogik (s.o.) angegeben.
- Urlaubslogiken: Hier werden nicht periodische Logiken für (Betriebs-) Urlaubszeiten und Feiertage hinterlegt. Die Wertetabelle ist eine Auflistung arbeitsfreier Tage (Datum, Monat, Jahr).

Definieren von Arbeitszeitmodellen (Abbildung 56-C)

Ein Arbeitszeitmodell ist stets durch je eine Pausen-, Wochen- und Urlaubslogik definiert. Es beschreibt für einen Zeitraum die geltenden Arbeitszeiten mit allen Arten von Unterbrechungen, lässt jedoch nicht die Modellierung von Ausnahmeregelungen wie z.B. Mehrarbeitsstunden an einem bestimmten Arbeitstag zu.

Editieren der Schichtkalender (Abbildung 56-D)

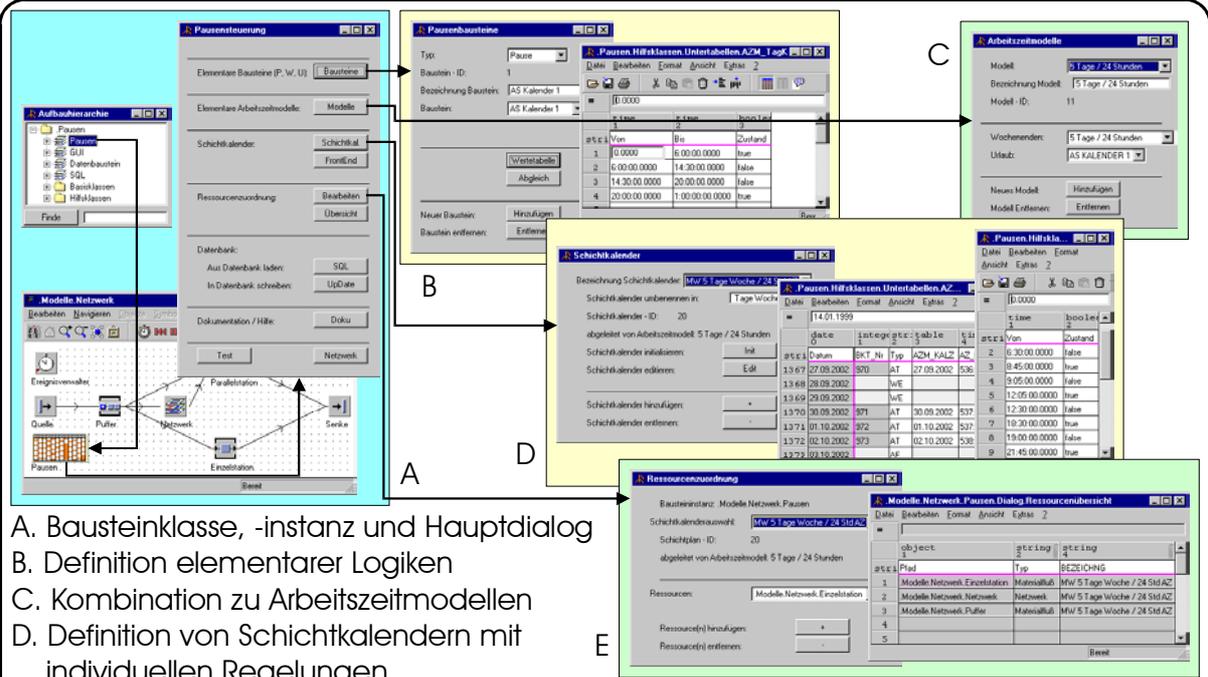
Ein Schichtkalender basiert jeweils auf einem Schichtmodell, bietet aber die Möglichkeit, abweichend von den auf elementaren Logiken basierenden Arbeitszeiten individuell abweichende Regelungen zu treffen, z.B. Sonderschichten an einzelnen Tagen. Die Wertetabelle enthält u.a. je Tag die Angabe des „Tagestyps“ (Arbeitstag, Wochenende, arbeitsfrei...) und bei Arbeitstagen eine Untertabelle mit den für genau diesen Tag geltenden Arbeits- und Pausenzeiten.

Einsatz in der Simulation

Nach einem Schichtkalender können beliebig viele Ressourcen eines Simulationsmodells arbeiten. Auch können verschiedene Schichtkalender unterschiedliche Ressourcen in einem Modell gleichzeitig steuern. Die Zuordnungen werden mit Hilfe der Objektauswahl-Methode des Basistoolsets zugewiesen und können, wie in Abbildung 56-E gezeigt, überprüft werden.

Drei Ressourcentypen können Schichtkalendern zugewiesen werden:

- eM-Plant Materialfluss-Grundbausteine: Der Pausenbaustein setzt während der Simulation bei jedem Pausenereignis des betreffenden Schichtkalenders das Attribut „Pause“ des Bausteins.
- Netzwerke: Wird ein Netzwerk einem Schichtkalender zugewiesen, so verhalten sich alle Materialfluss-Grundbausteine im angegebenen Netzwerk (und in darin liegenden Netzwerken) nach diesem Kalender.
- „Sonderbausteine“: Netzwerke, in denen je eine Methode mit dem Namen „Pause“ und „Entpausieren“ implementiert ist, werden als Sonderbausteine mit speziellen Verhaltensweisen bei Pausenereignissen erkannt. Diese Methoden werden vom Pausenbaustein entsprechend aktiviert.



A. Bausteinklasse, -instanz und Hauptdialog
 B. Definition elementarer Logiken
 C. Kombination zu Arbeitszeitmodellen
 D. Definition von Schichtkalendern mit individuellen Regelungen
 E. Zuordnen von Kalendern zu Simulationsressourcen

Pausenbaustein: Einsatz und Dialoge



Abbildung 56 - Pausenbaustein: Einsatz und Dialoge

Öffentliche Methoden

Die öffentlichen Methoden des Pausenbausteins umfassen u.a. Berechnungsmethoden, die z.B. zur Terminierung (2.4.1.1) eingesetzt werden. In Tabelle 7 sind die öffentlichen Methoden des Pausenbausteins kurz erklärt.

Methoden	Kurzbeschreibung
get_Rsc_Schichtplan	Ermitteln des Schichtplans, dem eine Ressource zugeordnet ist
set_Rsc_Schichtplan	Zuordnung einer Ressource zu einem Schichtkalender erstellen / entfernen
get_zug_Schichtplaen	Ermitteln der Schichtpläne, denen Ressourcen zugewiesen sind
ist_Arbeitstag	Stellt fest, ob ein Datum im angegebenen Schichtplans ein Arbeitstag ist
vorigerFreierTag	Ermitteln des einem Tag vorangehenden letzten arbeitsfreien Tages
nächsterFreierTag	Ermitteln des einem Tag folgenden nächsten Arbeitstages
vorigerArbeitstag	Ermitteln des einem Tag vorangehenden letzten Arbeitstages
nächsterArbeitstag	Ermitteln des einem Tag folgenden nächsten Arbeitstages
get_AZ_table	Ermitteln der summierten Arbeitszeit je Tag für ein Schichtmodell
holeAZDauer	Berechnen der in einem Intervall enthaltenen Arbeitszeit
holeAZTermin	Bestimmen des Bearbeitungsendes oder –beginns
holeBKT	Liefert für ein Zeitintervall die nach dem angegebenen Schichtmodell enthaltenen Arbeitstage
holeTermin	Liefert zu einem Start- oder Endtermin und gegebener Dauer den fehlenden Termin
holeAZbisPause	Ermitteln der verbleibenden Arbeitszeit bis zur nächsten Arbeitsunterbrechung
holeAZbisSchichtende	Ermitteln der verbleibenden Arbeitszeit bis zum Ende des Arbeitstages
holeAZbisUrlaub	Ermitteln der verbleibenden Arbeitszeit bis zum nächsten arbeitsfreien Tag

Tabelle 7 - Pausenbaustein: Öffentliche Methoden

2.2.6.2 Verfügbarkeit



Im Toolset Schiffbauliche Fertigung wurden u.a. diverse anzahlbehaftete Ressourcenklassen definiert, z.B. Werker, Werkzeuge (2.2.4.5.5 f.). Die Kapazitäten dieser Ressourcen können im Simulationsverlauf veränderlich sein, z.B. aufgrund von Fehlzeiten in Werkergruppen. Der Verfügbarkeitsbaustein bildet solches Verhalten ab und kann die Kapazität einer Ressource entsprechend steuern.

Damit der Verfügbarkeitsbaustein variierende zahlenmäßige Verfügbarkeit von Ressourcen steuern kann, muss:

- die Ressource eine Methode "verfuegb_anmelden" besitzen, um sich beim Verfügbarkeitsbaustein anzumelden (Bsp.: .verfuegb.Hilfsklassen.verfuegb_anmelden)
- die Ressource eine Variable "tab_verfuegb" besitzen, in der die Daten gespeichert werden können

- die Rsc_In-Schnittstelle der Ressource auf die Meldung "Verfügbarkeit update" reagieren können (s.a. .verfügbar.Hilfsklassen.Rsc_In)
- im Datenbaustein in der Tabelle Rsc_EIG_verfügbar der zeitliche Kapazitätsverlauf der Ressourcen eingetragen sein.

2.2.7 Auswertungstools

In diesem Abschnitt sind Bausteine beschrieben, die am Simulationsablauf nicht aktiv teilnehmen, sondern lediglich auf die Simulationsereignisse reagieren. Ihre Aufgabe ist hauptsächlich die statistische Auswertung und Visualisierung von Simulationsergebnissen. Die Eventschnittstelle bietet darüber hinaus die Möglichkeit, spezielle Logiken zu programmieren, um auf Simulationsereignisse zu reagieren.

2.2.7.1 Gantt



Der Gantt-Baustein dient der visuellen Darstellung von Ablaufplänen, die aus Planungsdaten, Simulations- oder Optimierungsläufen etc. stammen können. Basierend auf dem eM-Plant-Grundbaustein Gantt bietet er eine Reihe Zusatzfunktionen wie benutzerdefinierbare Darstellungsmodi, verschiedene Filter und ergänzende Informationen in Kopf- und Fußzeilen.

Aufbau und Funktion

Sofern Gantt-Charts unter Verwendung der vordefinierten Darstellungsmodi erstellt werden sollen, wird lediglich der Hauptdialog (Abbildung 57) benötigt. Die bisher implementierten Darstellungsmodi sind in Tabelle 8 aufgeführt. Abweichende Darstellungsformen, Filter usw. können in den unten beschriebenen weiterführenden Dialogen definiert werden.

Hauptmenü

Im oberen Pop-up-Menü wird ein Darstellungsmodus ausgewählt. Bei Bedarf kann dann ein Zeitfenster definiert werden, wenn nur die in einem bestimmten Zeitraum liegenden Vorgänge dargestellt werden sollen. Nach Betätigung des Buttons „Darstellen“ wird dann, sofern dem Gantt-Baustein Daten vorliegen, ein auf dem gewählten Darstellungsmodus basierendes Gantt-Chart generiert wie das in Abbildung 58 zu sehende Beispiel für eine Anwendung des Darstellungsmodus „Rahmentermine“.

Bei laufender Simulation werden die relevanten Ereignisse dem Gantt-Baustein durch Aufruf öffentlicher Methoden gemeldet und dort in Tabellen gespeichert. Es lassen sich aber auch Datenlagen darstellen, die im Datenkomplex „Arbeitspläne“ des Datenbausteins gespeichert sind. Hierfür ist der Button „Daten holen“ zu drücken, woraufhin die Daten in den Gantt-Baustein eingelesen werden.

Darstellungsmodi bearbeiten

Die Einstellungen, die einem Darstellungsmodus zugrunde liegen, lassen sich mit dem in Abbildung 59 gezeigten Dialog anzeigen und bearbeiten. Auch können neue Darstellungsmodi definiert werden. Der Dialog wird mit dem Button „Konfigurieren“ links oben im Hauptmenü aktiviert. Einige der Darstellungsmodi sind geschützt und können nicht editiert werden. Die Anzeige des Dialogs wird unterdrückt, wenn im Hauptdialog einer der geschützten Modi gewählt wurde.

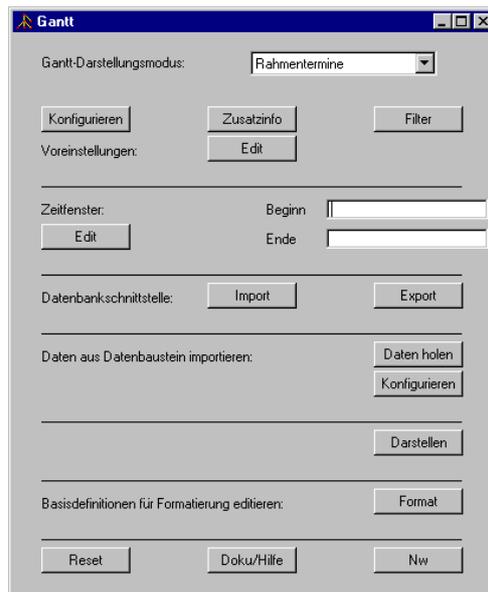


Abbildung 57 - Gantt: Hauptdialog

Modus	Kurzbeschreibung
Blockeinzelprozessablauf	Blockfertigung-Montage mit Eingangsbauteilen
Blockprozessablauf	Blockfertigung-Montage ohne Eingangsbauteile
Blöcke	
Ecktermine	Übersicht über die Rahmentermine von Schiffen und Bauabschnitten
Früh_SimF_SimS_Spät	3 Modi zur Gegenüberstellung von Planungs-, Simulations- und/oder Optimierungsdaten
Früh_Spät_Opt	
Früh_Spät_SimF_SimS_Opt	
Sektionsfertigung	
TestSim	Allgemeine Darstellungsform für Simulationsergebnisse

Tabelle 8 - Gantt: voreingestellte Darstellungsmodi

Der obere Teil des Dialogs enthält die grundlegenden Funktionen zum Anlegen, Umbenennen und Löschen eines Darstellungsmodus. In einem Eingabefeld kann eine kurze Beschreibung eingegeben werden. Mit dem Popup-Menü „Sortierung“ wird das Sortierverhalten beim Erstellen des Gantt-Charts festgelegt. Die Vorgänge können sortiert nach Start- oder Endterminen oder nach der Zusammenbaureihenfolge („Hierarchie of Assembly“) angeordnet werden.

Im mittleren Teil des Dialogs befinden sich die Elemente, mit denen auf drei Stufen festgelegt wird, welche Informationen darzustellen sind:

- **Key_Schedule:** Die erste Stufe betrifft die Auswahl der darzustellenden Vorgänge in Abhängigkeit von den Bauteilen bzw. Aufträgen, denen sie zugeordnet sind. Das hier maßgebende Attribut „Key_Schedule“ beschreibt den Rang eines Bauteils in der hierarchy of assembly. Mögliche Werte sind:
X (Endprodukt / Bauabschnitt), Y (taktbestimmende Baugruppe, i.d.R. Block / Modul), Z (Einzelteile, Komponenten)

- „Vorgangsarten“: In der zweiten Stufe werden für den gerade ausgewählten Key_Schedule-Wert die darzustellenden Vorgangsarten ausgewählt. Mögliche Werte sind:

Sammelvorgang, Bearbeitungsschritt, Montageschritt, AV, Pause, Störung, Blockierung, Unterprozess, Transport

- „Termine“: Im Prinzip kann es zu jeder Vorgangsart mehrere Terminpaare geben. Auf dieser dritten Stufe ist zu definieren, welche Termitypen für die Vorgänge anzuzeigen sind, die auf die gerade gewählte Kombination aus Key_Schedule und Vorgangsart anzuzeigen sind. Mögliche Werte sind:

Früheste, Späteste, Kann, Plan, Soll, Ist, Sim früh, Sim spät, Optimierte

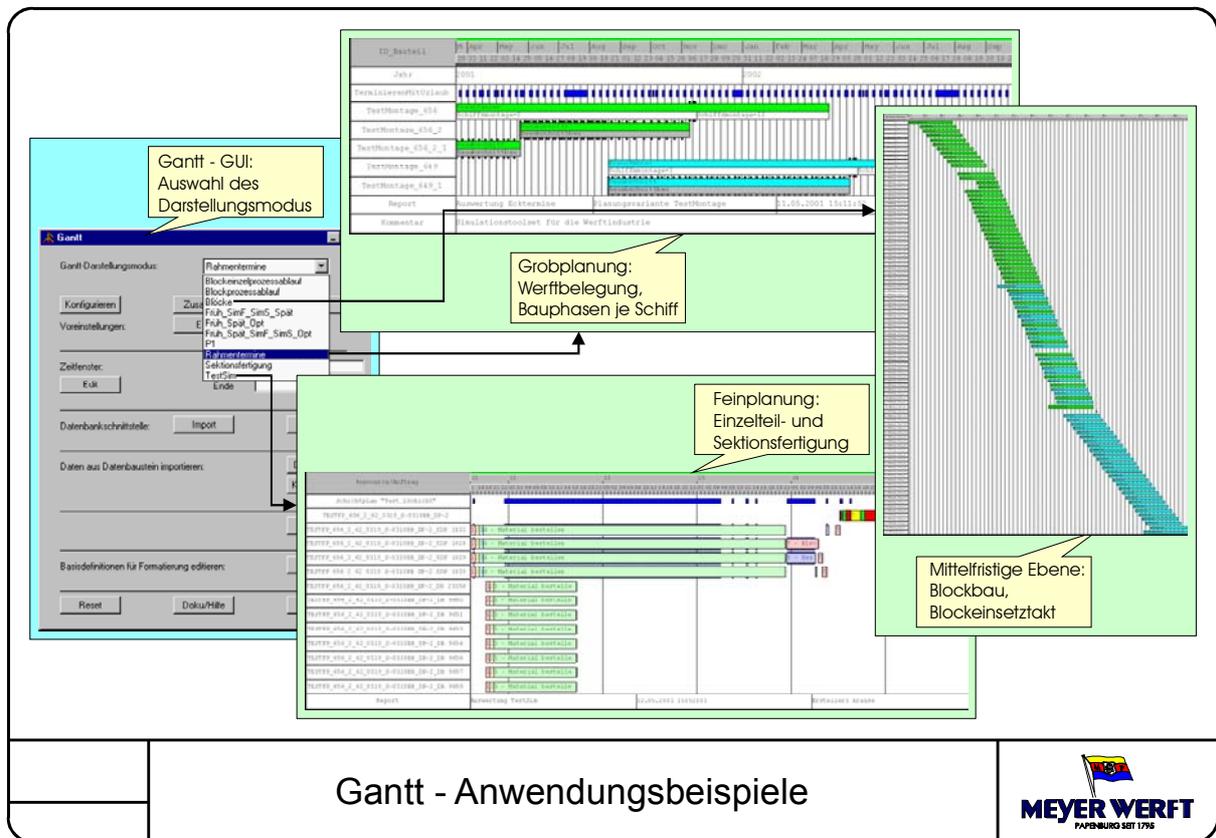


Abbildung 58 - Gantt: Beispiele verschiedener Darstellungsmodi

Der untere Teil des Dialogs enthält Elemente zur Festlegung der fünf Darstellungsparameter der Balken im Gantt-Diagramm für die gerade gewählte Kombination aus Key_Schedule, Vorgangsart und Termin. Für die Parameter Farbe, Muster, Höhe und Offset kann in einem separaten PopUp-Menü ein Modus für die Ermittlung der Parameterwerte der betreffenden Balken im Gantt-Chart gewählt werden. Möglich sind:

- „konstant“: allen betreffenden Balken im Chart wird der gleiche Parameterwert zugewiesen. Im rechten PopUp-Menü ist der gewünschte Wert zu wählen
- „Attribut“: um den Parameterwert zu bestimmen, wird ein zusätzliches Attribut ausgewählt. So kann z.B. erreicht werden, dass die Farbe des Balkens dem zugeordneten Kostenträger entspricht. Im rechten PopUp-Menü kann eines der für den jeweiligen Parameter verfügbaren Attribute ausgewählt werden.

- „verteilen“ (gilt nur für den Parameter „Offset“): Um zeitlich sich überlagernde Vorgänge, die ein bestimmtes Bauteil betreffen, übersichtlich darstellen zu können, sollte dieser Modus gewählt werden. Der Parameter Offset wird dann so angepasst, dass die Balken auf mehrere Zeilen verteilt werden.

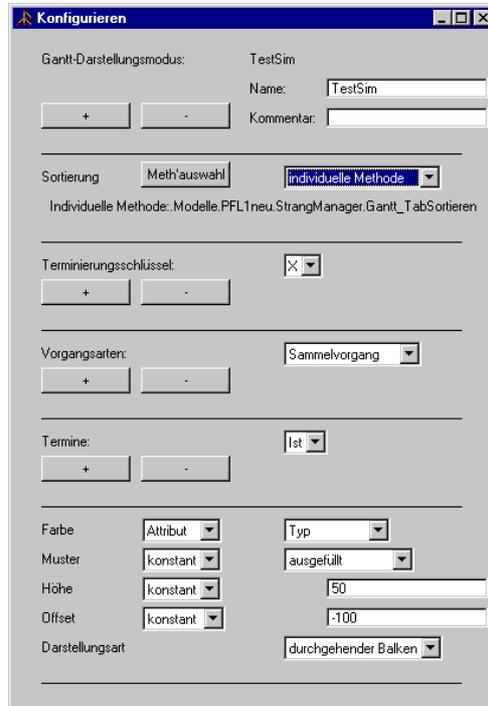


Abbildung 59 - Gantt: Darstellungsmodus konfigurieren

Der zusätzliche Parameter „Darstellungsart“ gibt an, ob die betreffenden Vorgänge mit einem Balken über die gesamte Vorgangsdauer dargestellt werden sollen, oder ob lediglich Vorgangsanfang und -ende mit einer senkrechten Linie angedeutet werden sollen.

Zusatzinformationen

Neben der Darstellung von Vorgängen können im Gantt-Chart noch ergänzende Informationen angezeigt werden. Mit dem Button „Zusatzinfo“ im Hauptdialog wird der in Abbildung 60 dargestellte Dialog aktiviert. Dort kann die Funktion generell aktiviert oder deaktiviert werden, außerdem kann im einzelnen angegeben werden, welche Zusatzinformationen angezeigt werden sollen. Folgende Zusatzinformationen sind verfügbar:

- Arbeitsfreie Tage, Pausen (nach Schichtkalender)
- Zeitskala (Jahresleiste, Monatsleiste, Wochenleiste, Tagesleiste)
- Darstellungsvariante
- Planungsvariante
- Erstellungsdatum
- Ersteller
- Kommentar

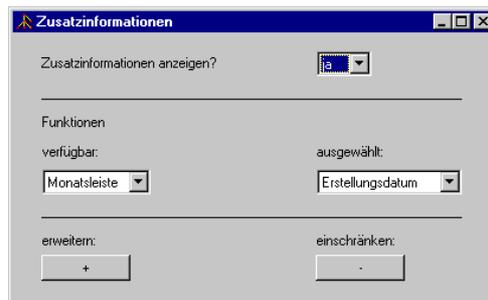


Abbildung 60 - Gantt: Auswahl Zusatzinformationen

Filter

Eine Methode, aus den vorliegenden Daten nur eine Teilmenge darzustellen, ist die Vorgabe eines Zeitfensters im Hauptdialog. Weitere Filter werden mit Hilfe des in Abbildung 61 gezeigten Dialoges definiert. Drei Filtertypen stehen zur Auswahl:

- Bauteil: Auswahl von Bauteilen, zu denen Vorgänge darzustellen sind. Wahlweise werden die Vorgänge zum ausgewählten Bauteil (Modus „self“) oder zum Bauteil und zu seinen Unterbauten gezeigt
- Ressource: Auswahl von Ressourcen, zu denen Vorgänge dargestellt werden sollen
- Vergabe: Im aktivierten Zustand werden wahlweise nur die Vorgänge zu Bauteilen dargestellt, die in der Produktstruktur einen Eintrag bei „Vergabe“ haben, bzw. nur die Vorgänge, die keinen entsprechenden Eintrag haben.

Der vierte Filter „Uhrzeit“ ist eigentlich kein Filter, sondern eine Sonderfunktion, die bewirkt, dass an jedem Balken im Diagramm die Start- und Endtermine auf ganze Tage abgerundet werden.

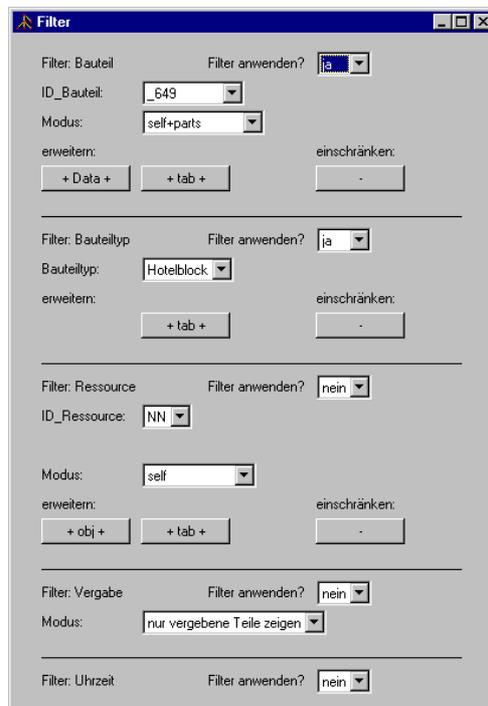


Abbildung 61 - Gantt: Definition von Filtern

2.2.7.2 BestandGantt



Dieser Baustein zeichnet den Belegungszustand eines Materialflussbausteins im Verlauf einer Simulation auf. Das Kapazitätsgebirge wird mit Hilfe des eM-Plant-Bausteins Gantt dargestellt.

Die Fachklasse des Bausteins wird im Simulationsmodell instanziiert und dann mit Hilfe des GUIs konfiguriert (Abbildung 62 - A, B). Obligatorisch ist eine zu beobachtende Ressource anzugeben (Materialflussbaustein). Optional können ein Zeitfenster für die Darstellung und ein auszuwertendes BE-Attribut für die differenzierte farbliche Darstellung der Auswertung angegeben werden.

Der Baustein arbeitet sowohl bei Vorwärts- als auch bei Rückwärtssimulation. In Abbildung 62 - C) wird beispielhaft ein Gantt-Chart mit einer Auswertung des BestandGantt-Bausteins gezeigt.

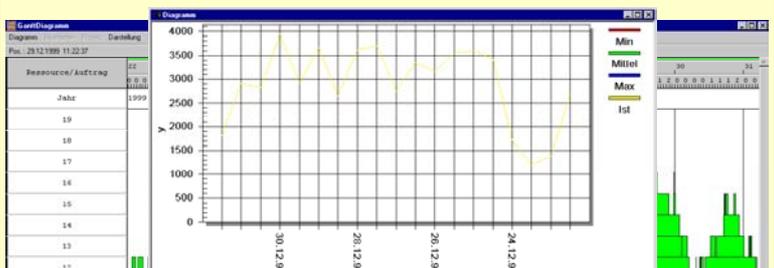
2.2.7.3 BestandDiagramm



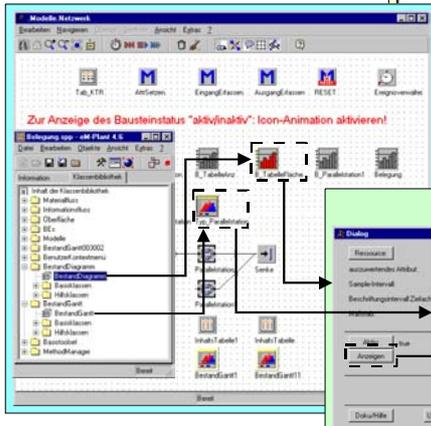
Dieser Baustein zeichnet den Belegungszustand eines Materialflussbausteins im Verlauf einer Simulation auf. Das Kapazitätsgebirge wird mit Hilfe des eM-Plant-Bausteins Diagramm dargestellt.

BestandDiagramm und BestandGantt unterscheiden sich vor allem in ihrer Darstellungsform. Aber auch die Funktionalitäten sind verschieden. BestandDiagramm arbeitet im „Sample“-Modus, d.h. der Belegungszustand einer Ressource wird in definierten Zeitabständen abgefragt.

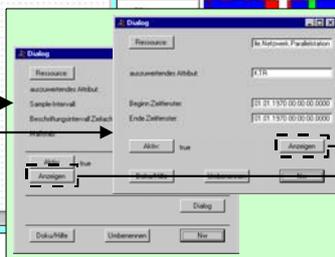
C. Auswertungen



A. Simulationsmodell mit Instanzen der Fachklassen



B. GUIs der Auswertungsbausteine



BestandGantt - BestandDiagramm



Abbildung 62 - BestandGantt / BestandDiagramm: Anwendungsbeispiel

Dieses Verhalten führt zu einer besseren Performance als der von BestandGantt praktizierte „Watch“-Modus, in dem die Ressource ständig überwacht und bei jedem Materialflussereignis eine Auswertung durchgeführt wird. Als Zusatzfunktion kann BestandDiagramm nicht nur die zum Auswertungszeitpunkt ermittelten aktuellen Belegungsstände darstellen, sondern auch Statistikwerte der Zeitintervalle (minimale / mittlere / maximale Belegung).

2.2.7.4 Eventschnittstelle



Die in Abbildung 63 gezeigte Eventschnittstelle ist das zentrale Element zur Annahme von Simulationsereignismeldungen. Sie verarbeitet die Meldungen lokal und / oder leitet sie an andere Bausteine (z.B. an Protokoll-, Auswertungs- oder Visualisierungskomponenten) weiter (siehe Abbildung 64).

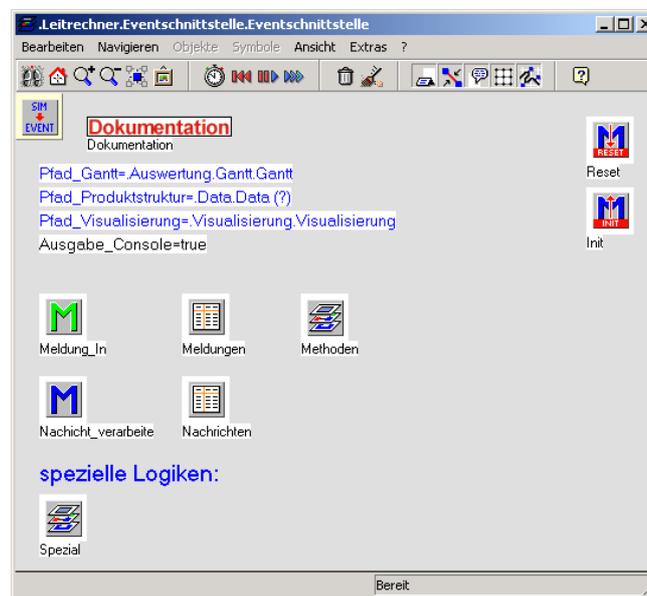


Abbildung 63 - Eventschnittstelle

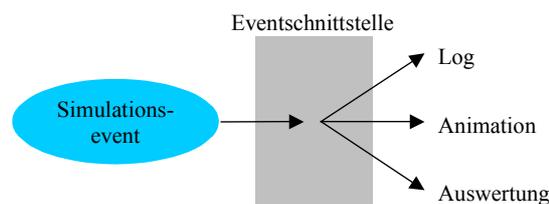


Abbildung 64 - Funktion Eventschnittstelle

Der Eventschnittstelle können unter anderem folgende Aktivitäten und Statusveränderungen gemeldet werden:

- Ressourcenanfrage, -reservierung, -buchung, -nutzung, -freigabe
- Ein- und Auslagervorgänge von BEs
- Verfahraktionen der beweglichen Maschine

Funktionen

Verarbeitung von Meldungen

Der Aufruf der Eventschnittstelle erfolgt über die öffentliche Methode „Meldung_In“. Ihre Parameter sind eine Meldung, die die Art des Ereignisses bezeichnet (z.B. „Bauteil erzeugt“), und ein Datenpaket (2.2.9.3). Im Datenpaket sind allgemeine (aufrufender Baustein) und spezifische Informationen, die von der jeweiligen Meldung abhängen, einzutragen.

Die von der Eventschnittstelle zu verarbeitenden Meldungen sind nicht festgelegt und können frei definiert werden. In der Tabelle „Meldungen“ ist für jede verarbeitbare Meldung eine auszuführende Methode hinterlegt. Im Netzwerk „Methoden“ werden die zu programmierenden Reaktionsmethoden modelliert.

Verarbeitung von Nachrichten

Durch Übermittlung der Meldung „Nachricht“ an die öffentliche Methode „Meldung_In“ kann die Eventschnittstelle auch genutzt werden, um Nachrichten in der eM-Plant-Konsole anzuzeigen. Im Datenpakete ist dann die Nachricht zu spezifizieren. In der Tabelle „Nachrichten“ werden die verarbeitbaren Nachrichten geführt. Jeder Nachricht ist ein Nachrichtentext zugeordnet, der in die Konsole zu schreiben ist. Der Text enthält Schlüsselwörter, die anhand des einrahmenden Sonderzeichens „&“ identifiziert sind. Im Datenpaket sind die Texte einzutragen, die anstelle der Schlüsselwörter in die Konsole zu schreiben sind.

Beispiel: Mit dem Datenpaket wird unter dem Index „Nachricht“ der Wert „Bauteil vernichtet“ übergeben. In der Tabelle „Nachrichten“ ist zu dieser Nachricht der Text „Bauteil &ID_Bauteil& wurde vernichtet“ gespeichert. Im Datenpaket steht beim Index „ID_Bauteil“ der Wert „Sektion0815“. In der Konsole erscheint die Nachricht „Bauteil Sektion0815 wurde vernichtet“.

2.2.8 Oberflächen



In diesem Abschnitt wird auf die Entwicklung von Komponenten eingegangen, die Simulationsexperten wie -laien benutzerfreundliche Oberflächen für den effektiven Einsatz des Simulationstoolsets ermöglichen. Anschließend werden Beispiele für den tatsächlichen Einsatz dieser Oberflächen gegeben.

Konzept

In der ersten Phase des Projekts konzentrierte sich die Arbeit auf die Erstellung von funktionalen Simulationsbausteinen. Damit der geplante Einsatz der Simulation auf der Meyerwerft realisiert werden kann, mussten diese Bausteine mit benutzerfreundlichen Oberflächen (Graphical User Interface GUI) ausgestattet werden. eM-Plant bietet dafür eine Schnittstelle an. Im Rahmen einer Diplomarbeit wurden ActiveX-Komponenten mit Visual Basic 5.0/6.0 erstellt und in die eM-Plant (Version 4.6.10) Umgebung eingebunden. Visual Basic wurde als Programmiersprache verwendet, da sowohl entsprechende Lizenzen als auch Erfahrungen im Umgang mit der Sprache innerhalb der Forschungsabteilung vorhanden sind (Werftstandard). Folgende Schwerpunkte wurden bearbeitet:

- Erstellung des Konzeptes für GUI im Bereich Simulation
- Erstellung von ActiveX-Steuerelementen als Ersatz für eM-Plant Dialogelemente
- Einbindung der GUI in eM-Plant
- Test der Softwarekomponenten
- Dokumentation der erstellten Software

Im Rahmen einer Diplomarbeit wurde ein Konzept entwickelt, mit dem einem Anwender durch die Einbringung von grafischen Oberflächen ein problemloses Arbeiten ermöglicht wird, ohne spezielle Kenntnis der Simulationssoftware vorauszusetzen.

Zusätzlich wurde aufgezeigt, wie die Kommunikation der GUI und systemeigener Objekte durchgeführt werden kann (Informationsaustausch zwischen einzelnen Objekten). Dank einer flexiblen Handhabung der Kommunikationsobjekte kann in verschiedenen Modellen bzw. Bausteinen eine Implementierung vorgenommen werden. Die erstellten Oberflächen können sowohl als eigenständige Anzeigeobjekte verwendet werden, als auch mittels einer erstellten Konzeption an einer Kommunikation teilnehmen.

1. Konzeption und Realisierung von GUI.
 - Erstellung der GUI mittels ActiveX-Technologie
 - Selbstständiges Registrieren der GUI in der Windows NT Registrierdatenbank
 - problemlose Implementierung in eM-Plant Modellen
 - Versionskontrolle verwendeter und erstellter ActiveX-Steuerelemente
 - Wiederverwendbarkeit aller ActiveX-Steuerelemente (GUI)
2. Konzeption und Realisierung der Kommunikation von GUI
 - Gewährleistung der Kommunikation der erstellten GUI
 - Erkennung aller kommunizierenden GUI
 - Flexibilität der Verwendung

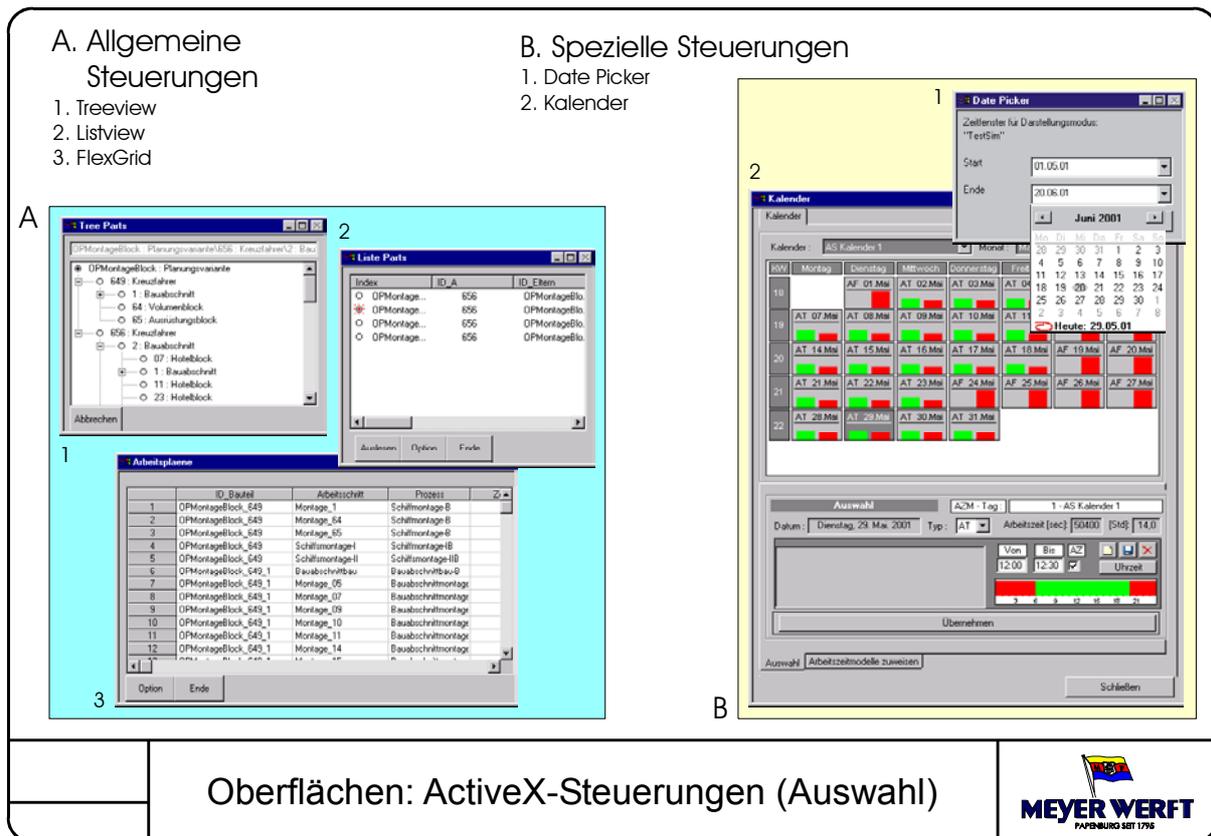
Für die Realisierung wurde die ActiveX-Schnittstelle von eM-Plant verwendet. Durch die Verwendung der ActiveX Technologie können zum einen sehr flexible GUIs erstellt werden, zum anderen kann bei der Erstellung der GUIs die gesamte Visual Basic Bandbreite von Steuerelementen verwendet werden.

Für den ersten Aufgabenbereich wurden die allgemein einsetzbaren ActiveX-Elemente „Treeview“ und „Listview“ erstellt. Zudem wurden spezielle ActiveX-Elemente entwickelt, die gemäß diesem Konzept als GUI in eM-Plant eingebracht wurden, um ausgewählte eM-Plant Dialogobjekte zu ersetzen. Gemäß der Aufgabenstellung wurden hier die grundlegenden Fragestellungen zum Einbringen von GUI in eM-Plant bearbeitet.

Für das Konzept Kommunikation der GUI wurden das ActiveX „Treeview“ sowie das ActiveX „Listview“ entsprechend verändert und das ActiveX „Flexgrid“ zusätzlich erstellt. Diese Komponenten sind vom Aufbau so gewählt, dass sie stellvertretend Anzeigeaufgaben für verschiedene Datenobjekte übernehmen, als auch an einer Kommunikation zwischen den ActiveX Elementen teilnehmen können.

Einsatz

Abbildung 1 zeigt einige der erstellten ActiveX-Steuerungen. Die allgemeinen Elemente Treeview, Listview und FlexGrid werden im dargestellten Einsatzfall zur Anzeige von Informationen aus dem Datenbaustein bzw. der Produktstruktur verwendet. Die speziell entwickelten Elemente finden Verwendung als Datums-Eingabeaufforderung bzw. als GUI für den Pausenbaustein.



Oberflächen: ActiveX-Steuerungen (Auswahl)

Abbildung 65 - Oberflächen: ActiveX-Steuerungen (Auswahl)

Die Implementierung zeigte die Leistungsfähigkeit der erstellten Software und verdeutlicht die Notwendigkeit der entwickelten Oberflächen. Mit Hilfe des Konzeptes für GUI wurden die grundlegenden Aufgabenstellungen der Implementierung, Versionssicherheit, Wiederverwendbarkeit und Flexibilität bearbeitet und umgesetzt. Die dabei entwickelten grafischen Oberflächen dienen der Anzeige lokal bestehender Datentabellen oder ersetzen eM-Plant Dialog-Elemente.

Die Erstellung getrennter Bausteine sowie die Trennung der Methoden-Rümpfe kommunizierender ActiveX-Elemente gemäß der Konzeption der Kommunikation von GUI, ermöglicht ein problemloses Einbringen der GUI in einem vom Benutzer zusammengestellten Simulationsmodell. Alle erstellten Kommunikations-GUI wurden so konzipiert, dass sie sowohl als eigenständige Oberfläche fungieren, als auch an einer Kommunikation teilnehmen können. Der erstellte GUI- sowie der Kommunikationsbaustein sind beliebig erweiterbar und sehr flexibel einsetzbar. Vorausgesetzt, entsprechende GUIs wurden in ein Modell implementiert, ist es dem Anwender nun möglich, Simulationsergebnisse, die in Tabellen innerhalb bestimmter Bausteine abgelegt wurden, mit Hilfe der kommunizierenden Oberflächen, problemlos anzeigen zu lassen.

Durch den selbstständigen Aufbau der Kommunikationstabelle gemäß des Konzeptes melden sich nur vorgesehene Kommunikationsobjekte dort an, sodass gewährleistet ist, dass nicht kommunikationsbereite Objekte beim Programmstart unberücksichtigt bleiben und somit Inkonsistenzen der zentralen Kommunikationstabelle vermieden werden.

Mittels der Realisierung des Gesamtkonzeptes sowie dessen Umsetzung und Aufbau in verschiedenen Modellen ist es nun möglich, das Simulationstoolset Anwendern zur Verfügung zu stellen, die die Struktur der Daten innerhalb eines Modells nicht kennen. Es können alle

gewünschten Daten des Toolsets durch einen Anwender gesichtet werden, insofern entsprechende GUI stellvertretend Anzeigoptionen übernehmen.

2.2.9 Kommunikation

Die Kommunikationselemente des Toolsets dienen dem Datenaustausch zwischen den Ressourcen, Steuerungskomponenten und Managern innerhalb der Simulationsumgebung.

Die Kommunikationselemente des Toolsets dienen dem Datenaustausch zwischen den Ressourcen, Steuerungskomponenten und Managern innerhalb der Simulationsumgebung.

2.2.9.1 Datenpaket

Die Schnittstelle zwischen dem Leitreechner, den Managern und den Ressourcenbausteinen muss eine möglichst hohe Flexibilität aufweisen, ohne an Kompatibilität zu verlieren. Dazu wurde ein Datenpaketprotokoll entworfen. Komplexe Anfragen oder Befehle können über standardisierte Datenpakete abgewickelt werden (es besteht weiterhin die Möglichkeit die Toolsetelemente über ihre öffentlichen Methoden anzusprechen).

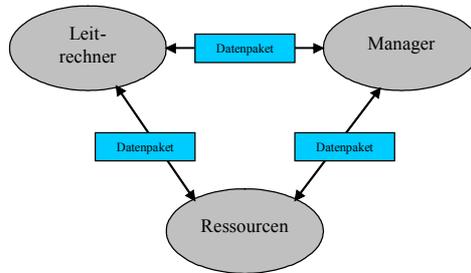
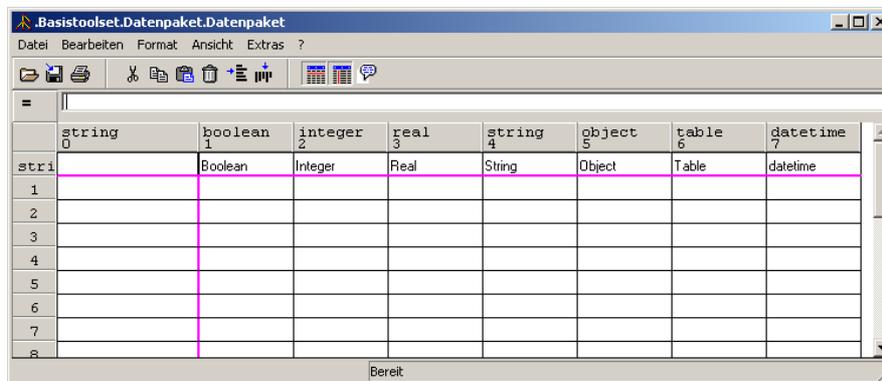


Abbildung 66 - Datenaustausch über Datenpakete

Abbildung 71 zeigt das im Ordner „Basistoolset“ als Tabelle implementierte Datenpaket. Die Anzahl und das Format der Spalten sind festgelegt. Für jeden Datentyp ist, direkt oder indirekt, eine Spalte vorgesehen. In der Zeilenindexspalte ist der jeweilige Zugriffsschlüssel einzutragen. Pro Zeile kann nur ein Parameter übergeben werden (Veraltet aber teilweise noch verwendet: Keine Zugriffsschlüssel und keine Beschränkung auf einen Eintrag pro Zeile). Komplexere Daten können in der frei formatierbaren „Tabelle“-Spalte übermittelt werden.

Sollen z.B. zwei Strings übergeben werden, so werden diese untereinander in der Spalte „String“ eingetragen und mit einem Zugriffsschlüssel versehen.



	0 string	1 boolean	2 integer	3 real	4 string	5 object	6 table	7 datetime
stri		Boolean	Integer	Real	String	Object	Table	datetime
1								
2								
3								
4								
5								
6								
7								
8								

Abbildung 67 - Datenpaket

Jedes Datenpaket wird mit einer als String formatierten Meldung an den Adressaten übergeben um den Inhalt des Datenpakets einordnen zu können.

Datenpakete können von allen Methoden des Leitrechners, der Manager und der Ressourcen abgesetzt werden. Alle Datenpakete werden zunächst der Methode „Kom_In“ des instanziierten Netzwerks „Kommunikation“ übergeben (siehe folgender Abschnitt). Dieses leitet das Datenpaket und die Meldung anschließend an die zuständige Methode bzw. den zuständigen Manager weiter.

Öffentliche Methoden

Bei der Kommunikation mit Hilfe von Datenpaketen sind folgende Methoden von Nutzen:

.Basistoolset.Datenpaket

Diese Methode gibt eine leere Datenpaket-Tabelle zurück, die anschließend mit Einträgen gefüllt und zur Kommunikation mit anderen Bausteinen verwendet werden kann.

.Basistoolset.F.F.Datenpaket_Eintrage

Diese Methode erzeugt einen Eintrag in einem Datenpaket. Als Parameter benötigt sie ein Datenpaket (*table*), den Index des zu erzeugenden Eintrags und die Bezeichnung des Datentyps des Eintrags (beide *string*) und den einzutragenden Wert (*boolean*, *integer*, *real*, *string*, *object*, *table* oder *datetime*).

2.2.9.2 Kommunikationsbaustein



Das in Abbildung 72 dargestellte Netzwerk „Kommunikation“ ist die Schnittstelle der Toolsetelemente zum Empfang von Datenpaketen.

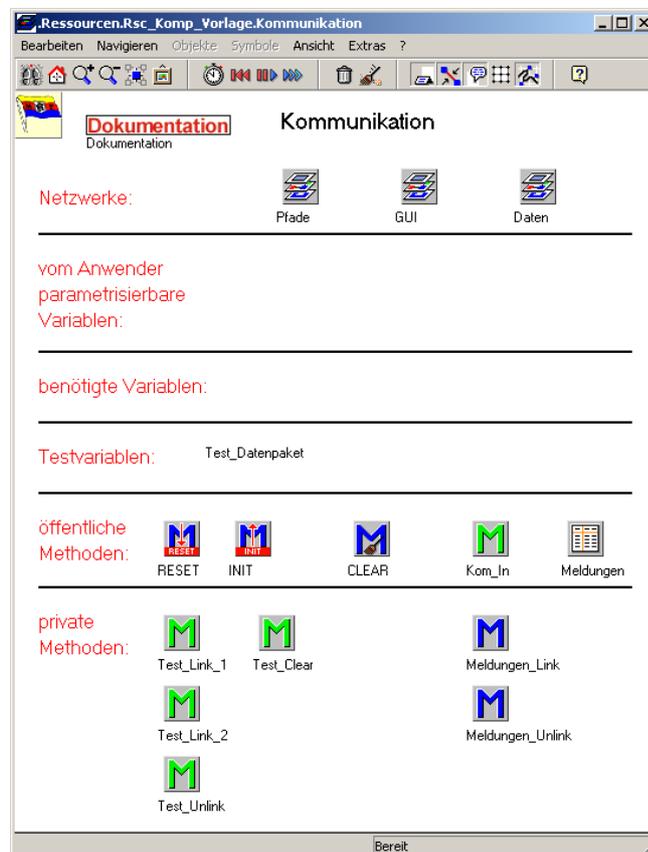


Abbildung 68 - Kommunikationskomponente

G:\Abschluss\Bericht\Gesamtbericht v03.11.doc

2.2.9.2.1 Schnittstellen

Öffentliche Methoden

Kom_In

Über die Methode werden Meldungen mit Datenpaketen an den Baustein übergeben. Die Tabelle „Meldungen“ enthält die jeweilige Reaktion des Bausteins auf den Befehl. Die folgenden Reaktionen kann ein Objekt auf einen Befehl zeigen:

- Ausführen einer Methode und Weitergabe des Datenpakets.
- Keine Reaktion falls der Befehl nicht unterstützt wird.
- Aufrufen des Debuggers falls der Befehl nicht unterstützt wird.

Verarbeitbare Meldungen

Meldungen Link

In jedem Kommunikationsbaustein werden nur Nachrichten hinterlegt, die durch eine lokale Methode oder Komponente verarbeitet werden können. Die Komponenten melden dazu ihre verarbeitbaren Nachrichten während der Instanzierung (Konstruktor-Methode) an den Baustein über die Meldung „Meldungen Link“.

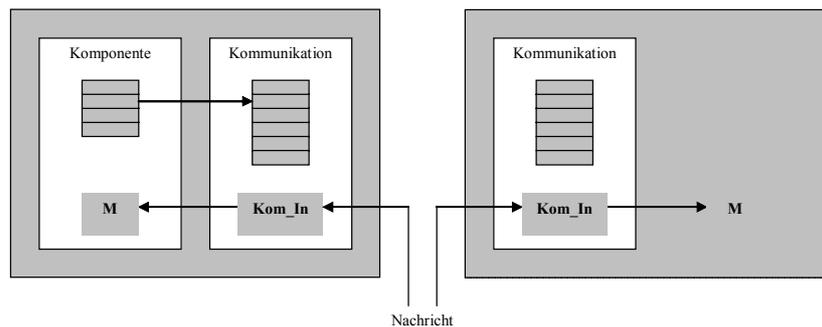


Abbildung 69 - Meldungsverarbeitung durch den Kommunikationsbaustein

Meldungen die in lokalen Methoden verarbeitet werden (die nicht von Komponenten angenommen werden), sind manuell in der Schnittstelle eingetragen.

Zeile	String	Object
1	An die Komponente weiterzuleitende Meldung 1	Komponente
...	...	
n	An die Komponente weiterzuleitende Meldung n	

Tabelle 9 - Meldungen Kommunikationsbaustein: Meldungen Link

Meldungen Unlink

Zeile	String	Object
1	Meldung 1 die nicht weiter an die Komponente weiterzuleiten ist	Komponente
...	...	

n | Meldung n die nicht weiter an die Komponente weiterzuleiten ist |

Tabelle 10 - Meldungen Kommunikationsbaustein: Meldungen Unlink

2.2.9.3 Datenpaket

Die Schnittstelle zwischen dem Leitrechner, den Managern und den Ressourcenbausteinen muss eine möglichst hohe Flexibilität aufweisen, ohne an Kompatibilität zu verlieren. Dazu wurde ein Datenpaketprotokoll entworfen. Komplexe Anfragen oder Befehle können über standardisierte Datenpakete abgewickelt werden (es besteht weiterhin die Möglichkeit die Toolsetelemente über ihre öffentlichen Methoden anzusprechen).

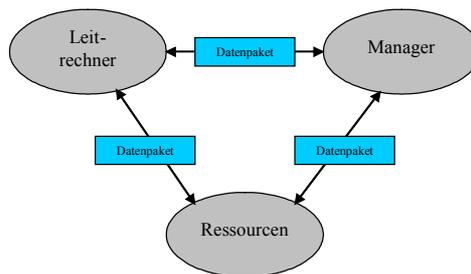
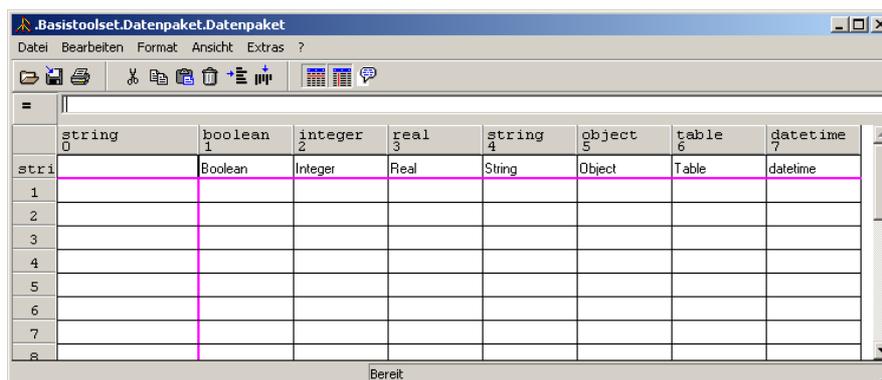


Abbildung 70 - Kommunikation: Datenaustausch über Datenpakete

Abbildung 71 zeigt das im Ordner „Basistoolset“ als Tabelle implementierte Datenpaket. Die Anzahl und das Format der Spalten sind festgelegt. Für jeden Datentyp ist, direkt oder indirekt, eine Spalte vorgesehen. In der Zeilenindexspalte ist der jeweilige Zugriffsschlüssel einzutragen. Pro Zeile kann nur ein Parameter übergeben werden (Veraltet aber teilweise noch verwendet: Keine Zugriffsschlüssel und keine Beschränkung auf einen Eintrag pro Zeile). Komplexere Daten können in der frei formatierbaren „Tabelle“-Spalte übermittelt werden.

Sollen z.B. zwei Strings übergeben werden, so werden diese untereinander in der Spalte „String“ eingetragen und mit einem Zugriffsschlüssel versehen.



	string 0	boolean 1	integer 2	real 3	string 4	object 5	table 6	datetime 7
stri		Boolean	Integer	Real	String	Object	Table	datetime
1								
2								
3								
4								
5								
6								
7								
8								

Abbildung 71 - Kommunikation: Datenpaket

Jedes Datenpaket wird mit einer als String formatierten Meldung an den Adressaten übergeben um den Inhalt des Datenpakets einordnen zu können.

Datenpakete können von allen Methoden des Leitrechners, der Manager und der Ressourcen abgesetzt werden. Alle Datenpakete werden zunächst der Methode „Kom_In“ des instanziierten Netzwerks „Kommunikation“ übergeben (siehe folgender Abschnitt). Dieses leitet das

Datenpaket und die Meldung anschließend an die zuständige Methode bzw. den zuständigen Manager weiter.

Öffentliche Methoden

Bei der Kommunikation mit Hilfe von Datenpaketen sind folgende Methoden von Nutzen:

.Basistoolset.Datenpaket

Diese Methode gibt eine leere Datenpaket-Tabelle zurück, die anschließend mit Einträgen gefüllt und zur Kommunikation mit anderen Bausteinen verwendet werden kann.

.Basistoolset.F.F.Datenpaket_Eintrage

Diese Methode erzeugt einen Eintrag in einem Datenpaket. Als Parameter benötigt sie ein Datenpaket (*table*), den Index des zu erzeugenden Eintrags und die Bezeichnung des Datentyps des Eintrags (beide *string*) und den einzutragenden Wert. (*boolean*, *integer*, *real*, *string*, *object*, *table* oder *datetime*).

2.2.9.4 Kommunikationsbaustein



Das in Abbildung 72 dargestellte Netzwerk „Kommunikation“ ist die Schnittstelle der Toolsetelemente zum Empfang von Datenpaketen.

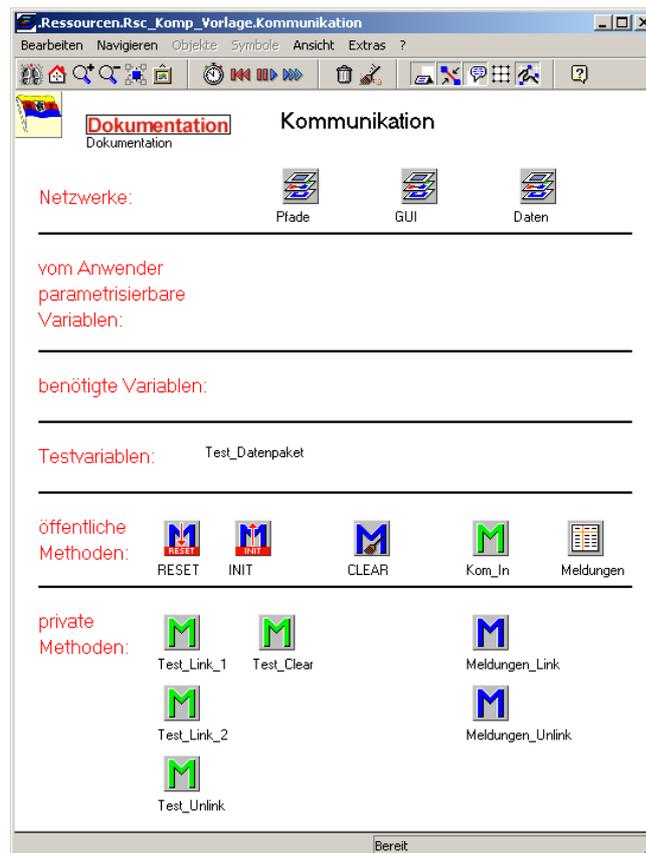


Abbildung 72 - Kommunikation: Kommunikationskomponente

2.2.9.4.1 Schnittstellen

Öffentliche Methoden

Kom_In

Über die Methode werden Meldungen mit Datenpaketen an den Baustein übergeben. Die Tabelle „Meldungen“ enthält die jeweilige Reaktion des Bausteins auf den Befehl. Die folgenden Reaktionen kann ein Objekt auf einen Befehl zeigen:

- Ausführen einer Methode und Weitergabe des Datenpakets.
- Keine Reaktion falls der Befehl nicht unterstützt wird.
- Aufrufen des Debuggers falls der Befehl nicht unterstützt wird.

Verarbeitbare Meldungen

Meldungen Link

In jedem Kommunikationsbaustein werden nur Nachrichten hinterlegt, die durch eine lokale Methode oder Komponente verarbeitet werden können. Die Komponenten melden dazu ihre verarbeitbaren Nachrichten während der Instanzierung (Konstruktor-Methode) an den Baustein über die Meldung „Meldungen Link“.

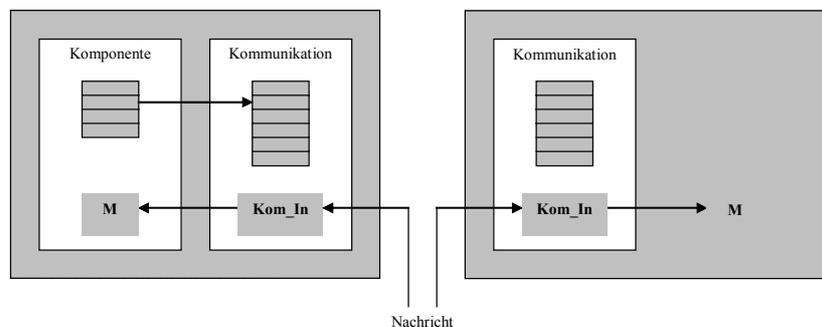


Abbildung 73 - Meldungsverarbeitung durch den Kommunikationsbaustein

Meldungen die in lokalen Methoden verarbeitet werden (die nicht von Komponenten angenommen werden), sind manuell in der Schnittstelle eingetragen.

Zeile	String	Object
1	An die Komponente weiterzuleitende Meldung 1	Komponente
...	...	
N	An die Komponente weiterzuleitende Meldung n	

Tabelle 11 - Meldungen Kommunikationsbaustein: Meldungen Link

Meldungen Unlink

Zeile	String	Object
1	Meldung 1 die nicht weiter an die Komponente weiterzuleiten ist	Komponente
...	...	
N	Meldung n die nicht weiter an die Komponente weiterzuleiten ist	

Tabelle 12 - Meldungen Kommunikationsbaustein: Meldungen Unlink

2.3 Datenmanagement

Für den geplanten Einsatz der Simulation auf Planungs- und Meisterebene musste im Umfeld der eigentlichen Simulationsanwendung einige Voraussetzungen geschaffen werden. Mit der Entwicklung des integrierten Datenmodells musste eine leistungsstarke „Front End“ Applikation entwickelt werden, mit der das gesamte Datenmodell bearbeitet werden kann.

2.3.1 Architektur

Basis für die Simulation bildet das entwickelte und auf dem SQL Server abgelegte Datenmodell. Alle Applikationen werden so zentral aus dem Datenmodell mit Simulationsdaten versorgt. Diese Daten werden teilweise aus Fremdsystemen (CAD, Ablaufplanung) übernommen bzw. exklusiv für die Simulation erzeugt. Für die Datenpflege und Datenerzeugung musste eine Umgebung („Front End“ Applikation) entwickelt werden, mit der der Anwender diese Simulationsaufgabe bewältigen kann. Die Architektur (Client Server) und das Datenmodell wurden bereits in 2.1 beschrieben.

2.3.2 SDM Tool

Für die Datenpflege wurde die „Front End“ Applikation SDM (Simulation Daten Management) entwickelt. Mit dem SDM Tool können die Bereiche Produkt, Arbeitsplan und Ressourcen des Datenmodells bearbeitet werden. Funktionen zur Auswertung von Simulationsexperimenten wurden ebenfalls im SDM Tool umgesetzt. Ein weiterer sehr wichtiger Bereich ist die Verwaltung der Anwender und die Verwaltung von Berechtigungen innerhalb des SDM Tools. Bei der Entwicklung des Tools mussten die Anforderungen an Interaktivität und Anwenderfreundlichkeit unbedingt berücksichtigt werden. Das SDM Tool wurde während des gesamten Projektes kontinuierlich weiterentwickelt (Release zum Projektabschluss 7.1.9)

2.3.2.1 Berechtigungen und Verwaltung

Im SDM Tool wurde ein Berechtigungskonzept integriert. Über das Berechtigungskonzept kann das Menü und der zur Verfügung gestellte Funktionsumfang für jeden Anwender gesteuert werden. Durch diese Funktion kann dem Anwender eine auf seine individuellen Bedürfnisse abgestimmte Anwendung zur Verfügung gestellt werden.

Die Anwender können in drei Ebenen eingeteilt werden. Auf der obersten Ebene steht die Gruppe der Administratoren. Dieser Gruppe steht der volle Funktionsumfang des SDM Tools zur Verfügung. In der mittleren Ebene befindet sich die Gruppe der Planer und Meister. Planern kann ein individuelles Menü zur Verfügung gestellt werden. Generell stehen dieser Gruppe jedoch nicht die Funktionen für die Verwaltungsaufgaben zur Verfügung. Auf den ersten beiden Ebenen kann das Datenmodell vollständig und interaktiv vom Anwender bearbeitet werden. Anwender auf der untersten Ebene (Viewer) haben die Möglichkeit, sich verschiedene Datenmodelle (Produkt, Arbeitsplan, Ressourcen) und verschiedene Auswertungen anzeigen zu lassen. Die Modellierung von Daten wurde auf dieser Ebene vollständig unterdrückt.

Durch das Berechtigungskonzept kann die SDM Anwendung sehr flexibel eingesetzt und individuell konfiguriert werden.

Folgende Verwaltungsfunktionen wurden innerhalb des SDM Tool integriert:

- Anmeldung und Passwortabfrage
- Verwaltung der Login und Logouts

- Verwaltung der Simulationsvarianten.
- Verwaltung der Anwender und der Berechtigungen
- Verwaltung der SDM-Funktion (Menü)
- Konfiguration des SDM Tools

2.3.2.2 Verwalten von Varianten

Eine Anforderung an das Datenmanagement ist es, eine projektübergreifende Simulation zu ermöglichen. Dazu werden verschiedene Kostenträger in Varianten verwaltet und organisiert. In einer Variante kann somit das Produktionsspektrum eines größeren Zeitraums abgebildet werden. Varianten können exklusiv für einen Anwender oder für Anwendergruppen (Multiuser) freigeschaltet werden. In einer Variante können mehrere Planer mit unterschiedlichen Planungsaufgaben gleichzeitig arbeiten. Im SDM Tool wird der Zustand von Varianten (Multiuser oder Singleuser) angezeigt und verwaltet. Gleichzeitig wird angezeigt, von welchem Planer die verschiedenen Varianten benutzt werden.

2.3.2.3 Datenmodell im SDM Tool

Für die Pflege der drei großen Bereiche des Datenmodells Produkt, Arbeitsplan und Ressourcen wurden im SDM Funktionen erstellt.

Produkt:

Für den Bereich des Produktmodells wurde der Komplex „Bauplan“ entwickelt. In diesem Komplex wurden alle Funktionen, die zur Bearbeitung des Produktes erforderlich sind, gebündelt. Folgende Funktionen wurden entwickelt:

- Pflege der Hierarchie of Assembly

In der Hierarchie of Assembly wird modelliert, wie und in welchen Stufen das Schiff (Endprodukt) gebaut wird. Dabei muss eine Modellierung der Daten aus Sicht der Simulation ermöglicht werden. Einzelne Bauteile (Zwischenprodukte wie Blöcke und Sektionen) können mit den Funktionen Gruppieren, Zuordnen und Teilen interaktiv modelliert werden.

- Kopier- und Löschfunktion

Für die Datenpflege wurden verschiedene Kopierfunktionen hinterlegt. Mit diesen Funktionen können ganze Varianten, aber auch einzelne Schiffe kopiert werden. Mit den Löschfunktionen können ganze Varianten, einzelne Schiffe, aber auch einzelne Blöcke und Sektionen gelöscht werden. Die Pflege des Datenmodells wird mit diesen Funktion erheblich vereinfacht.

- Pflege von Attributen

Für die Pflege von einzelnen Attributen wurden die entsprechenden Eingabemöglichkeiten geschaffen.

- Berechnung und Bearbeitung des Geometriemodells

Das Acht-Punkte-Geometriemodell kann jederzeit neu berechnet werden. Diese Funktion ist erforderlich, wenn Veränderungen in der Hierarchie off Assembly vorgenommen wurden.

- 3D Visualisierung des Produktes auf Basis des Geometriemodells

Für die Visualisierung der Schiffe wurde eine 3D Visualisierung des Produktes in das SDM Tool integriert.

Zentrales Element in diesem Komplex ist die Visualisierung der Hierarchie of Assembly. Abbildung 74 zeigt eine Übersicht zu den Funktionen des Komplexes Bauplan im SDM Tool.

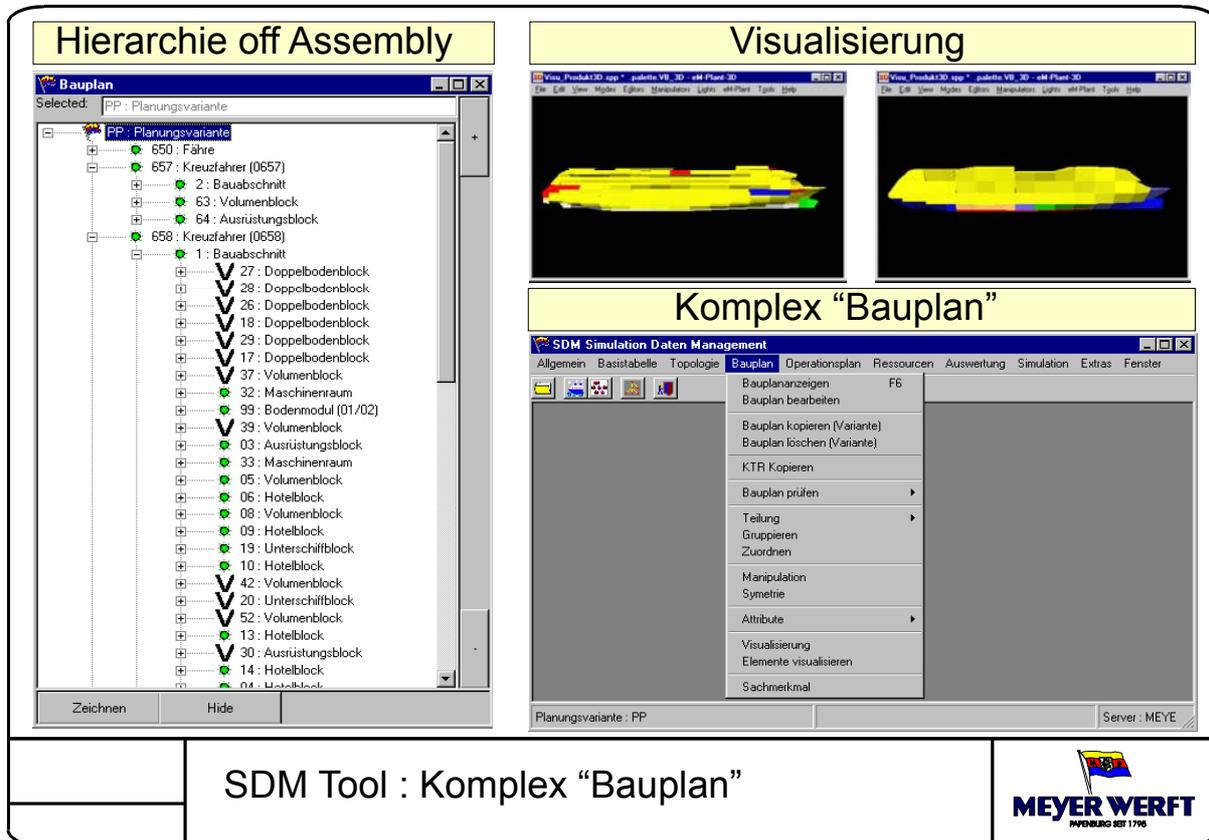


Abbildung 74 - SDM Tool: Komplex „Bauplan“

Arbeitspläne:

Für den Bereich des Arbeitsplanmodells wurde der Komplex „Operationsplan“ entwickelt. In diesem Komplex wurden alle Funktionen gebündelt, die zur Bearbeitung der Arbeitspläne erforderlich sind. Folgende Funktionen wurden entwickelt:

- Modellierung von Arbeitsplänen
In den Arbeitsplänen werden die einzelnen Fertigungsschritte modelliert. Im Arbeitsplan sind somit für jeden Arbeitsschritt die Fertigungsdauer und die Termine hinterlegt.
- Modellierung von Anordnungsbeziehungen
Anordnungsbeziehungen verknüpfen die einzelnen Arbeitsschritte miteinander.
- Zuordnung von Ressourcen

Im Arbeitsplan werden die einzelnen Arbeitsschritte mit den Ressourcen verknüpft. Die Zuordnung der Ressourcen zu den Arbeitsschritten erfolgt im Komplex Operationsplan.

- Erstellung von Fertigungsaufträgen

Für die Simulationsmodelle müssen Fertigungsaufträge erstellt werden. Fertigungsaufträge können individuell für die Simulation erzeugt werden. Verschiedene Filter und Auswahlmöglichkeiten stehen dem Anwender zur Verfügung.

- Lösch- und Kopierfunktionen und Arbeitsplangenerator

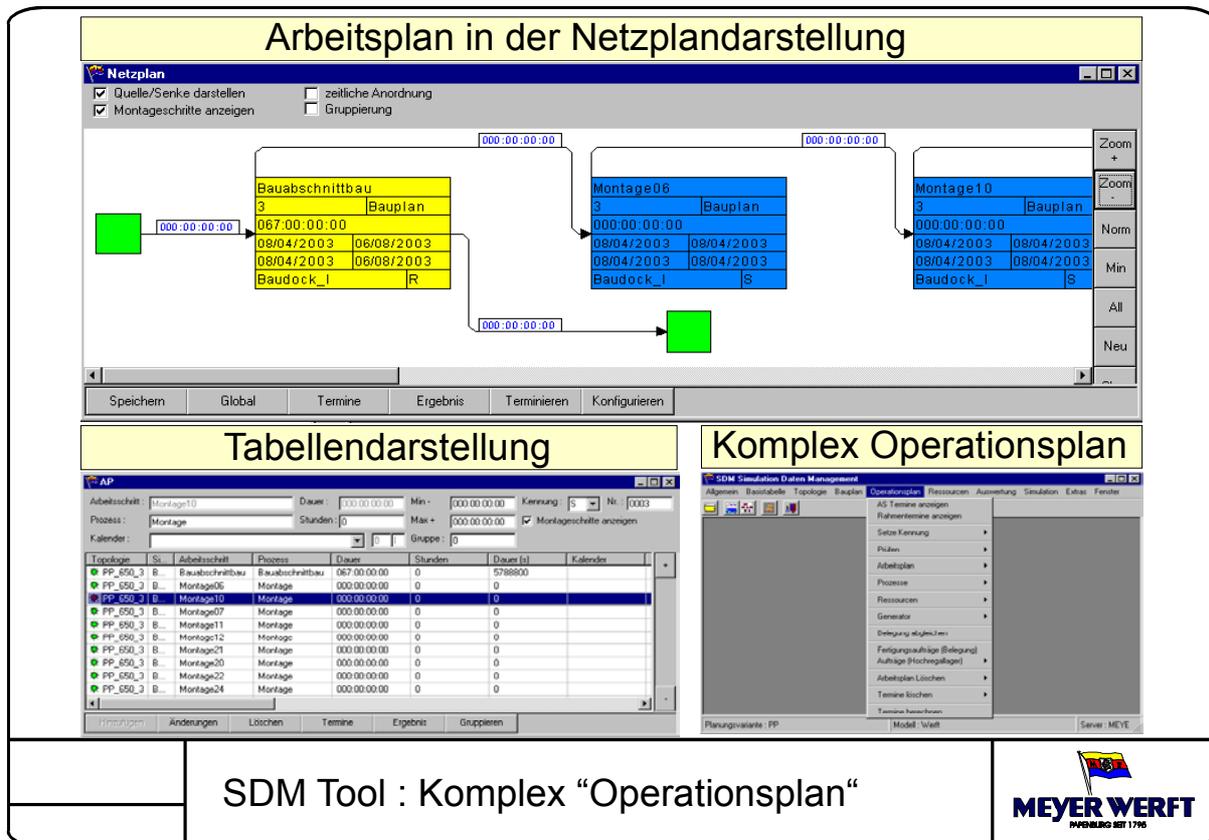
Für die Datenpflege wurden verschiedene Kopier- und Löschfunktion hinterlegt. Des weiteren wurde ein Arbeitsplangenerator in den Komplex integriert (vgl. 2.5.2).

- Übernahme von Terminen und Vorgängen aus der Ablaufplanung

Innerhalb des Komplexes Operationsplan wurde eine Schnittstelle zur Ablaufplanung integriert. Mit Hilfe der Schnittstelle können Termine und Vorgänge aus der Ablaufplanung übernommen werden.

- Funktionen zur Berechnung von Terminen

Termine für einzelne Arbeitsschritte können lokal berechnet werden. Eine Terminierung einer gesamten Variante erfolgt innerhalb der Simulationsanwendung Terminierung.



Arbeitsplan in der Netzplandarstellung

Netzplan

Quelle/Senke darstellen zeitliche Anordnung
Montageschritte anzeigen Gruppierung

000:00:00:00

Bauabschnittbau 3 Bauplan
067:00:00:00
08/04/2003 06/08/2003
Baudock_I R

000:00:00:00

Montage06 3 Bauplan
08/04/2003 08/04/2003
08/04/2003 08/04/2003
Baudock_I S

000:00:00:00

Montage10 3 Bauplan
08/04/2003 08/04/2003
08/04/2003 08/04/2003
Baudock_I S

Speichern Global Termine Ergebnis Terminieren Konfigurieren

Tabellendarstellung

AP

Arbeitsschritt: Montage10 Dauer: 000:00:00:00 Min: 000:00:00:00 Kennung: S Nr.: 0003
Prozess: Montage Stunden: 0 Max: 000:00:00:00 Montageschritte anzeigen
Kalender: [] Gruppe: []

Topologie	SI	Arbeitsschritt	Prozess	Dauer	Stunden	Dauer (s)	Kalender
PP_650_3 B	B	Bauabschnittbau	Bauabschnittbau	067:00:00:00	0	5788800	
PP_650_3 B	B	Montage06	Montage	000:00:00:00	0	0	
PP_650_3 B	B	Montage10	Montage	000:00:00:00	0	0	
PP_650_3 B	B	Montage07	Montage	000:00:00:00	0	0	
PP_650_3 B	B	Montage11	Montage	000:00:00:00	0	0	
PP_650_3 B	B	Montage12	Montage	000:00:00:00	0	0	
PP_650_3 B	B	Montage21	Montage	000:00:00:00	0	0	
PP_650_3 B	B	Montage20	Montage	000:00:00:00	0	0	
PP_650_3 B	B	Montage22	Montage	000:00:00:00	0	0	
PP_650_3 B	B	Montage24	Montage	000:00:00:00	0	0	

Änderungen Löschen Termine Ergebnis Gruppieren

Komplex Operationsplan

SDM Simulation Daten Management

Abgesehen Basistabelle Topologie Bauplan

Operativplan Ressourcen Auswertung Simulation Extras Fenster

- AS Termine anzeigen
- Rahmentermine anzeigen
- Setze Kennung
- Plänen
- Arbeitsplan
- Prozesse
- Ressourcen
- Generator
- Belegung abfragen
- Fertigungsaufträge (Belegung)
- Aufträge (Hochregallage)
- Arbeitsplan Löschen
- Termine Löschen
- Termine bearbeiten
- Modell: Welt
- Server: MEYE

Planungsvariante: PP

SDM Tool : Komplex "Operationsplan"



Abbildung 75 - SDM-Tool: Komplex Operationsplan

Zentrales Element in diesem Komplex ist die Visualisierung der Arbeitsschritte in einem interaktiven Netzplan. Abbildung 75 zeigt eine Übersicht zu den Funktionen des Komplexes Operationsplan im SDM Tool.

Ressourcen:

Für den Bereich des Ressourcenmodells wurde der Komplex „Ressourcen“ entwickelt. In diesem Komplex wurden alle Funktionen die zur Bearbeitung der Ressourcen erforderlich sind gebündelt. Folgende Funktionen wurden entwickelt:

- Gruppieren von Ressourcen in Modellen
Ressourcen werden in Modellen für die Simulation gruppiert
- Verwaltung der allgemeinen und der speziellen Attribute
Attribute wie Abmessungen der Ressource, Zuordnung der Ressource zu einem Schichtkalender etc. müssen modelliert werden.
- Anlegen und Löschen von Ressourcen
Ressourcen müssen angelegt, verändert und auch gelöscht werden

Zentrales Element in diesem Komplex ist die Visualisierung der Ressourcen in einem Modell. Abbildung 76 zeigt eine Übersicht zu den Funktionen des Komplexes Ressourcen im SDM Tool.

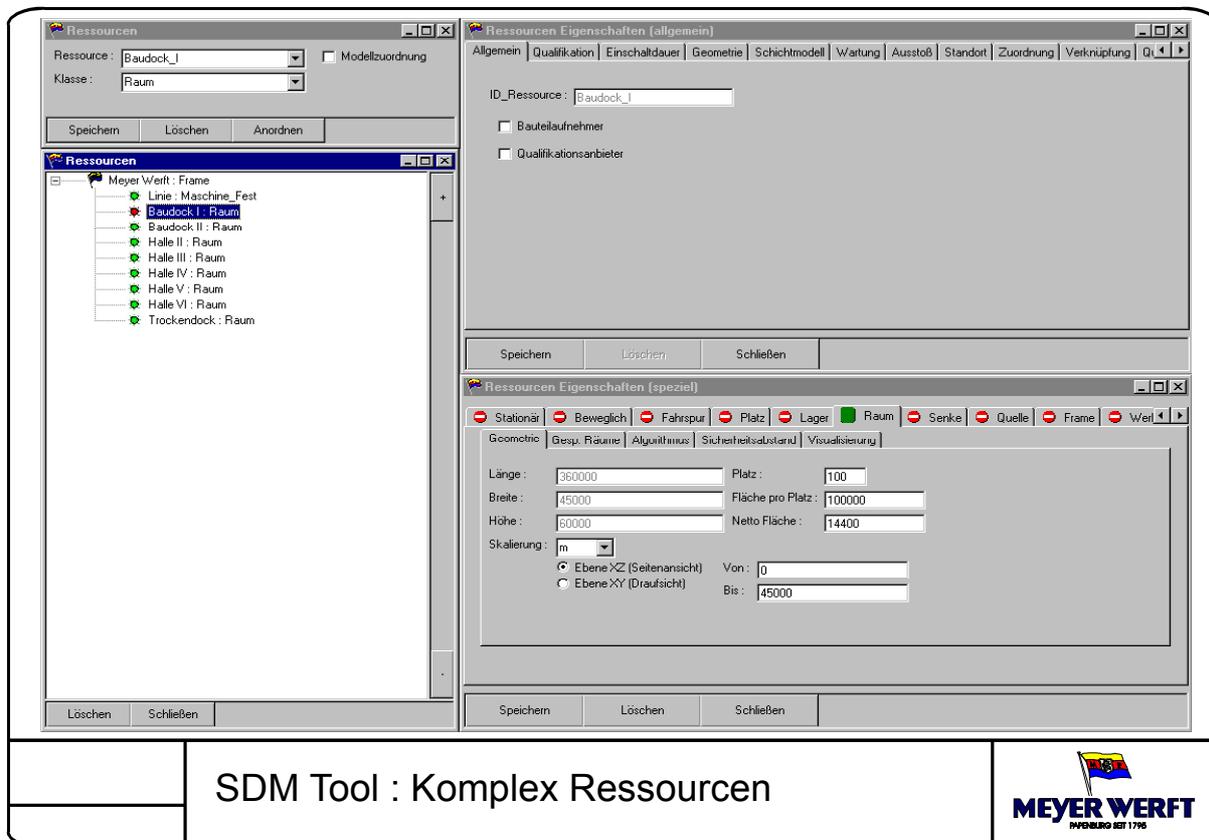


Abbildung 76 - SDM Tool: Komplex „Operationsplan“

Ressourcendaten sind Stammdaten und werden bei der Fertigungssimulation nicht oft verändert. Produktdaten und Arbeitsplandaten werden sehr häufig für die einzelnen Simulationsläufe verändert. Änderungen von Daten in einem Komplex wie z.B. Veränderung der Baustrategie haben auch Änderungen im Komplex der Operationspläne zur Folge. Im SDM Tool werden diese Änderungen in einem Komplex erkannt und automatisiert, in den anderen Komplexen verändert.

Datenmodellprüfung

Die eigentlichen Simulationsmodelle arbeiten auf der Basis des integrierten Datenmodells. Damit es nicht zu Fehlern in den Modellen kommt (Abbruch der Simulation), muss die Konsistenz des Datenmodells gewährleistet werden. Für die Prüfung des Datenmodells wurden in allen drei Komplexen Prüfroutinen hinterlegt, mit denen die Konsistenz innerhalb des Komplexes, aber auch die Konsistenz des Gesamtmodells geprüft werden kann. Diese Funktionen sind ein sehr wichtiges Werkzeug für den Planer. Die geforderten und integrierten Routinen wurden in der Pilotphase entwickelt.

2.3.2.4 Auswertung

Auswertungen von Simulationsexperimenten wurden im SDM Tool integriert. Auswertungen können in zwei Gruppen klassifiziert werden. In der ersten Gruppe sind alle allgemeinen Auswertungen wie Gantt- und Kapazitätsdiagramme zusammengefasst. In der zweiten Gruppe wurden die speziellen Auswertungen, wie z.B. Visualisierung der Flächenbelegung, zusammengefasst.

Gantt- und Kapazitätsdiagramme

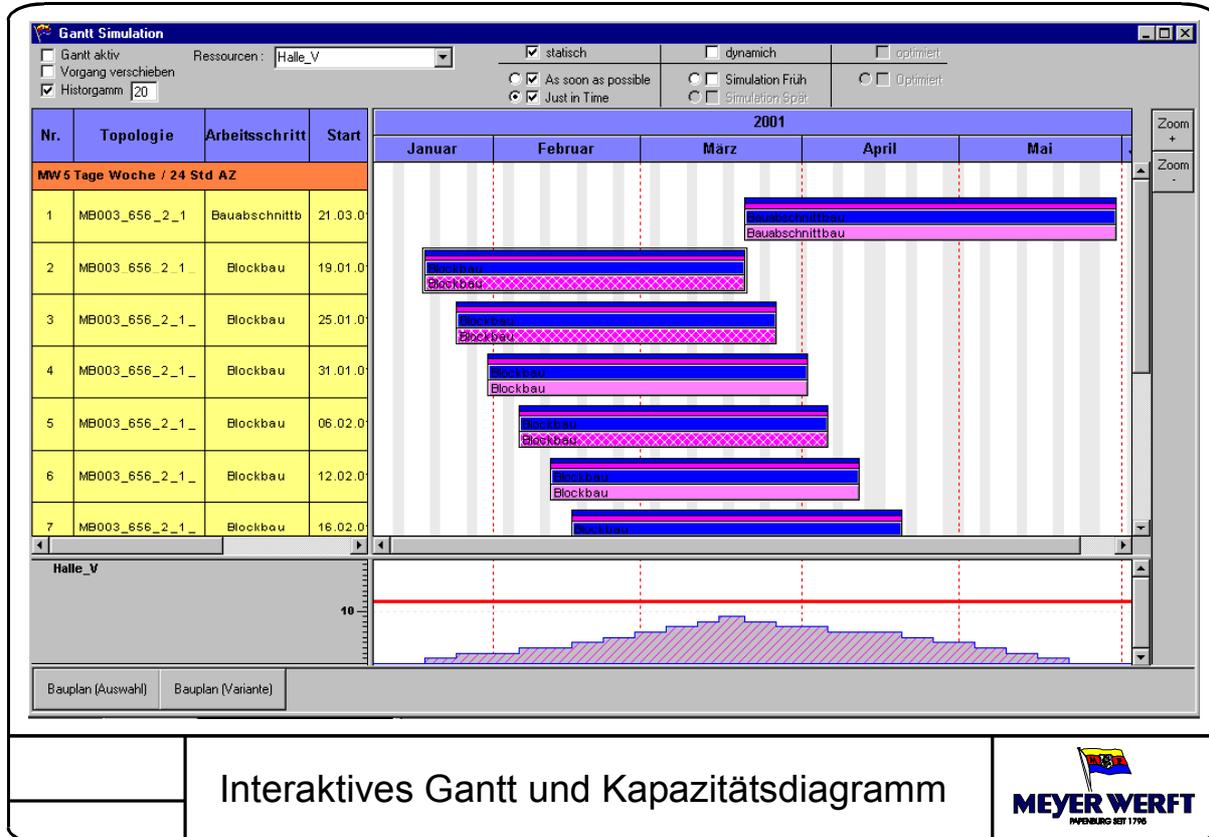


Abbildung 77 - SDM Tool: Interaktives Gantt- und Kapazitätsdiagramm

Alle Planungs- und Simulationstermine können im Gantt-Diagramm ausgewertet werden. Dem Planer stehen verschiedene Filter und Auswahlmöglichkeiten zur Verfügung. Das Gantt-Diagramm ist interaktiv aufgebaut, so dass es auch zur Modellierung genutzt werden kann. Verschiedene Auswertungen können in allgemeinen Kapazitätsdiagrammen angezeigt werden. Abbildung 77 zeigt das Gantt- und das Kapazitätsdiagramm.

Visualisierung der Raumbelegung

Für die Auswertung der Simulation der Raumbelegung wurde eine spezielle interaktive Plantafel entwickelt. Von der Simulation wird ein Vorschlag für eine Raumbelegung erzeugt. Dennoch ist es erforderlich, dass eine solche Raumbelungsvariante vom Planer bearbeitet und modifiziert werden kann. Dies erfolgt im SDM Tool mit Hilfe der interaktiven Plantafel. Die Plantafel wurde auf dem integrierten Datenmodell aufgesetzt und arbeitet somit mit dem selben Datenstamm wie alle Simulationsanwendungen.

Die Visualisierung von Räumen (Flächen) kann in der Seitenansicht und in der Draufsicht erfolgen. Die Auswahl der Visualisierungsart erfolgt durch den Planer. Für den Bereich der Docks ist die Seitenansicht sinnvoll. Alle anderen Hallen werden in der Draufsicht dargestellt. Mit der Plantafel wird aufgezeigt, wo und wie lange ein Bauteil (Schiff, Block, Sektion) einen Teil eines Raumes belegt. Die Ansicht kann wahlweise, je nach Planungshorizont, auf Tagesansicht, Wochenansicht und Monatsansicht geschaltet werden.

Mit der Plantafel kann in der Seitenansicht und in der Draufsicht vollständig interaktiv gearbeitet werden. Die Auswahl der einzelnen Elemente erfolgt mit Hilfe der Hierarchy of Assembly. Das Positionieren der Bauteile in den jeweiligen Hallen wird dann mit Hilfe von Drag-and-Drop-Funktionen durchgeführt. Abbildung 78 zeigt die Plantafel.

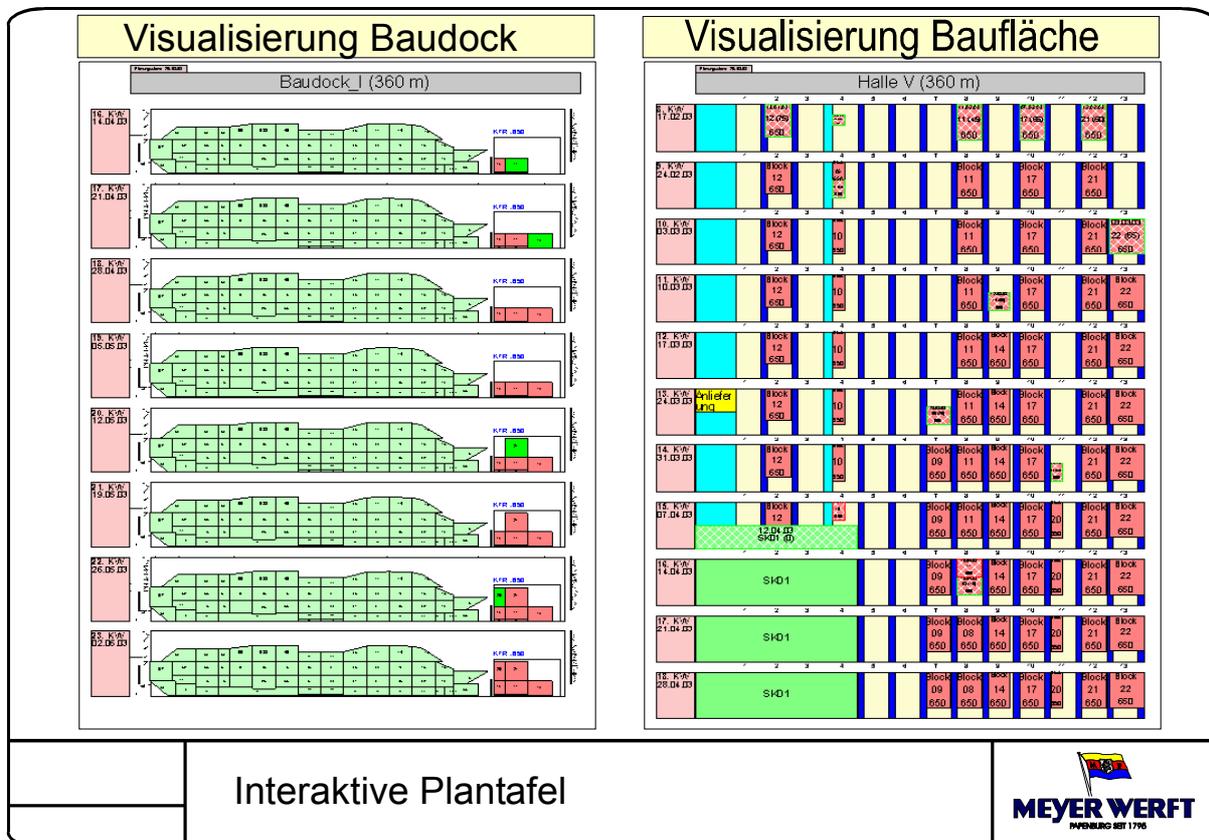


Abbildung 78 - SDM Tool: Interaktive Plantafel

Problemfenster

Problemfenster werden verwendet, um den Planer gezielt an Probleme wie z.B. Terminverschiebungen und Terminverletzungen heranzuführen. Mit diesen Auswertungen kann der Planer das Simulationsergebnis automatisiert prüfen. Probleme können so schnell gefunden und vom Planer bewertet werden. Ein manuelles Durchsuchen der Simulationsergebnisse ist somit nicht erforderlich.

2.3.3 Modellverwaltung

Modelle, die auf der Basis des Simulationstoolsets entwickelt wurden, müssen dem Anwender (Planer, Meister) zur Verfügung gestellt werden. Dabei kann bzw. muss das Simulationsmodell mit unterschiedlichen Rechten gestartet werden. So erlaubt die Entwicklungslizenz den Vollzugriff auf alle Tabellen, Methoden etc., die Anwendungslizenz hingegen startet ein Simulationsmodell in einem gesicherten Zustand (nicht editierbare Tabellen und Methoden etc.). Diese Funktionalität wurde auf der Seite der Anwendungslizenz um verschlüsselte Methoden ergänzt. Durch die Implementierung eines Rechte- und Berechtigungskonzeptes wird für den Simulationsexperten das Modell mit einer Entwicklungslizenz und für den Anwender mit einer Anwendungslizenz gestartet. Modelle können so vom Anwender nicht verändert bzw. beschädigt werden.

Des weiteren wird beim Start eines Modells eine Modelloberfläche aktiviert, mit welcher der jeweilige Anwender das Modell steuern kann. Dies wurde auf der Simulationsebene durch die Verwendung von „AutoexecMeth“ (vgl. 2.2.2) realisiert.

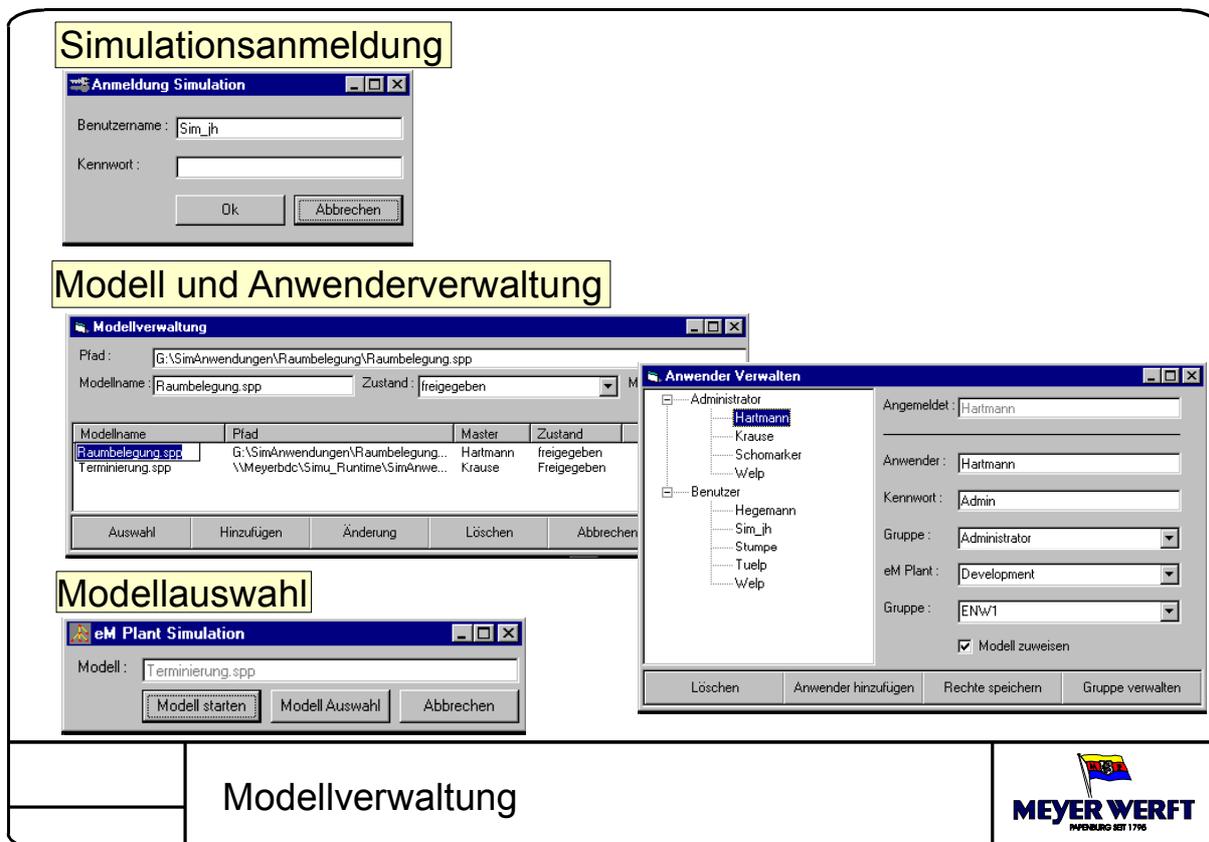


Abbildung 79 - Modellverwaltung

Dem Anwender muss für seine Planungs- bzw. Simulationenaufgabe ein individuelles Portfolio von Modellen bereitgestellt werden. Modelle, die vom Simulationsexperten gepflegt und gewartet werden, müssen für diesen Zeitraum für den Endanwender gesperrt werden.

Für die Anmeldung wurde eine Modell- und Anwenderverwaltung aufgebaut, mit der verschiedenen Anwendern Modelle zugewiesen und Berechtigungen erteilt werden können. Die Modellverwaltung wurde in das Toolset integriert und Anwendern zur Verfügung gestellt. Abbildung 79 zeigt die Modell- und Anwenderverwaltung. Der Start der Modellverwaltung kann direkt aus Windows, beziehungsweise indirekt aus dem SDM erfolgen. Beide Funktionen wurden im Projekt realisiert.

2.3.4 Piloteinsatz

Der Piloteinsatz der gesamten Simulationslösung war für die Entwicklung des Toolsets sehr wichtig. Durch die aktive Einbindung von Simulationsanwendern konnten die Anforderungen und Erweiterungen, die sich aus diesem Piloteinsatz ergaben, in der Entwicklung des Toolsets berücksichtigt werden.

Zur Piloteinführung der Simulation auf Planer- und Meisterebene (Arbeitsvorbereitung) wurde im zweiten Halbjahr 2001 mit ersten Schulungsmaßnahmen begonnen. Die Schulungen umfassten die Themen "Modellierung von Daten", "Arbeiten mit Modellen" und "Auswertung von Simulationsexperimenten".

Modellieren von Daten :

Das Datenmodell bildet die Grundlage für jede Simulationsanwendung im Forschungsprojekt. Daher lag bei der Einführung der Simulation der Schwerpunkt bei der Modellierung von Simulationsdaten. Die Schulung in diesem Bereich umfasst im wesentlichen die Einführung der Applikation "SDM". Die Schulung wurde in die Bereiche "Modellierung des Produktes", "Modellierung von Arbeitsplänen" und "Modellierung von Ressourcen" unterteilt.

Arbeiten mit Modellen:

Das Arbeiten mit Modellen umfasst die Bedienung von unterschiedlichen Simulationsmodellen. Die Auswahl und die Ausführung von Simulationen wird durch die Komponente "Modellverwaltung" gewährleistet, die jedem Anwender zur Verfügung gestellt wurde. Die einzelnen Modelle können dann durch individuelle Oberflächen gesteuert werden. Die Schulung in diesem Bereich umfasste im wesentlichen eine Einweisung in die Bedienung der einzelnen Modelle. Für die Modelle "Terminierung", "Flächenbelegung" und "TB-Simulation" wurden verschiedene Anwender geschult.

Auswertung von Simulationsexperimenten:

Simulationsergebnisse können im SDM (Visualisierung der Flächenbelegung, Kapazitätsdiagramme, Gantt etc.) ausgewertet werden. Die Schulungsmaßnahmen in diesem Bereich umfassten die Erläuterung der einzelnen Funktionen und wurden den Bedürfnissen der Anwender individuell angepasst.

Das Erreichen des Projektzieles "Einsatz der Simulation auf Planer und Meisterebene" konnte durch den Piloteinsatz aufgezeigt werden. Gleichzeitig waren die Rückmeldungen von Planern, Meistern etc. erforderlich, damit das Toolset im Projekt gezielt weiter entwickelt werden konnte.

2.4 Die Virtuelle Werft

Die virtuelle Werft muss thematisch in zwei Bereiche unterteilt werden. Der erste Bereich umfasst den Einsatz der Simulation in der Fertigungsplanung. Dies kann von der strategischen über die taktische bis hin zur operativen Ebene erfolgen. Der zweite Bereich umfasst die Fabrik- bzw. die Investitionsplanung. In beiden Bereichen sollen auf der Basis des Simulationstoolsets Simulationsmodelle erstellt werden, mit denen die jeweilige Planungs- bzw. Simulationaufgabe bearbeitet werden kann. Simulationen werden klassisch im ersten Bereich eingesetzt.

Der Einsatz der Simulation ist im Bereich der Fertigungsplanung auf der Anwenderebene geplant. Der Bereich der Investitionsplanung bleibt den Simulationsexperten vorbehalten. Für Simulationsstudien ist der folgende Ablauf (Abbildung 80) geplant.

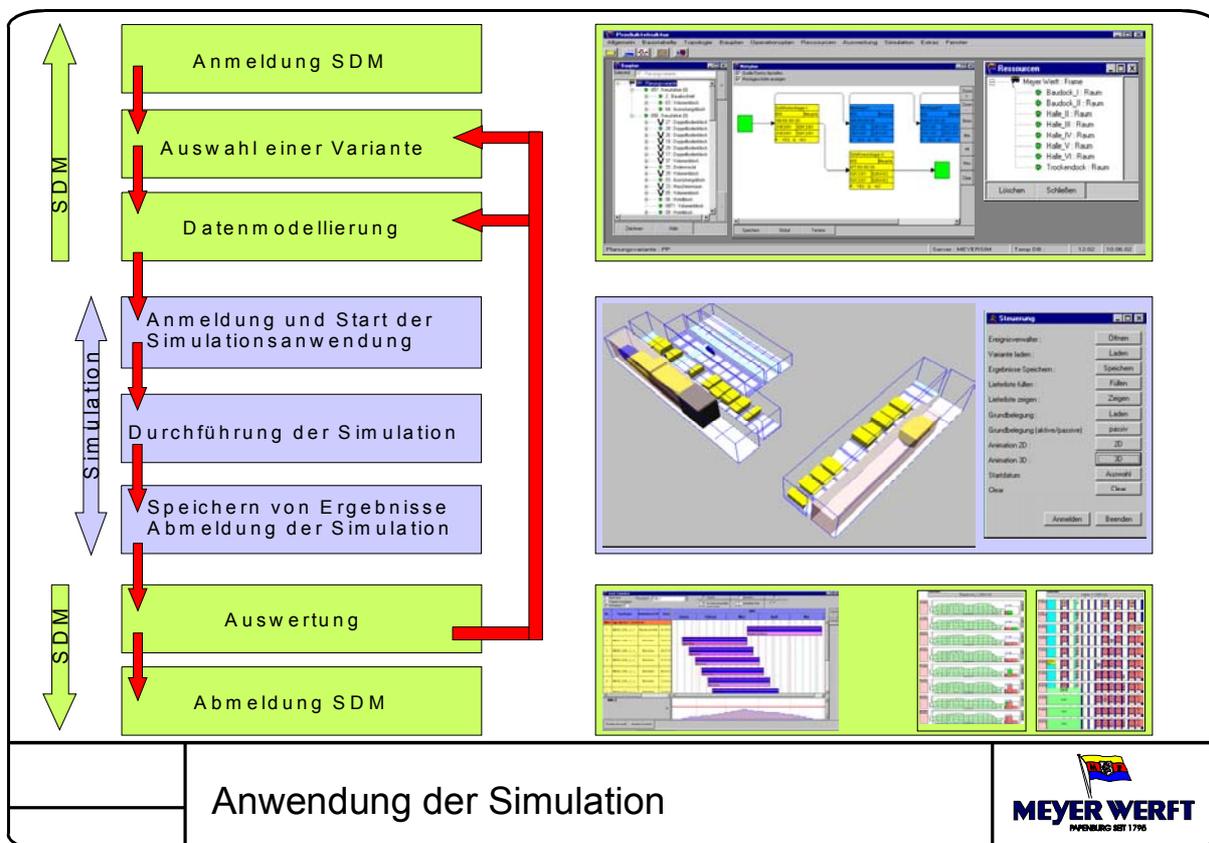


Abbildung 80 Virtuelle Werft: Workflow für Simulationsstudien

Im Projekt wurde der Einsatz der Simulation in der Pilotphase realisiert. Im Piloteinsatz sollten die entwickelten Konzepte und der Einsatz des gesamten Simulationstoolset getestet werden. Für die Pilotphase wurden in beiden Bereichen Simulationsmodelle erstellt, die den jeweiligen Anwender zur Verfügung gestellt wurden.

Eine Einbindung der Anwender war im gesamten Projekt sehr wichtig. Besonders das Zusammenspiel der einzelnen Werkzeuge musste intensiv getestet und stetig verbessert werden um die hoch gesteckten Projektziele (Interaktivität, Anwenderfreundlichkeit etc.) in diesem Bereich zu erreichen.

2.4.1 Fertigungsplanung (Anwendungsebene)

Für die Unterstützung der Fertigungsplanung wurden verschiedene Modelle aufgebaut und den Anwendern nach einer Schulungsphase zur Verfügung gestellt. Die Durchführung der Simulationsstudien lag im Bereich der Fertigungsplanung komplett im Verantwortungsbereich der Anwender (Planer).

2.4.1.1 Sim_Schedule



Das in eM-Plant realisierte Modell „Sim_Schedule“ ist ein Terminierungsmodul, das dazu verwendet wird, um die in einer Planungsvariante noch fehlenden Plantermine zu errechnen. Rückwärts- und Vorwärtsterminierung bestimmen die Spätesten und Frühesten Termine, eine zusätzliche Funktion ist die Terminierung der Topologiesichten übergreifenden, hierarchisch aufgebauten Netzpläne.

Das Modell kann vom SDM-Tool aus aktiviert werden. Es erscheint zunächst der abgebildete Dialog (Abbildung 81). Er bietet die elementaren Funktionen der Terminierung an, die im folgenden erläutert werden.

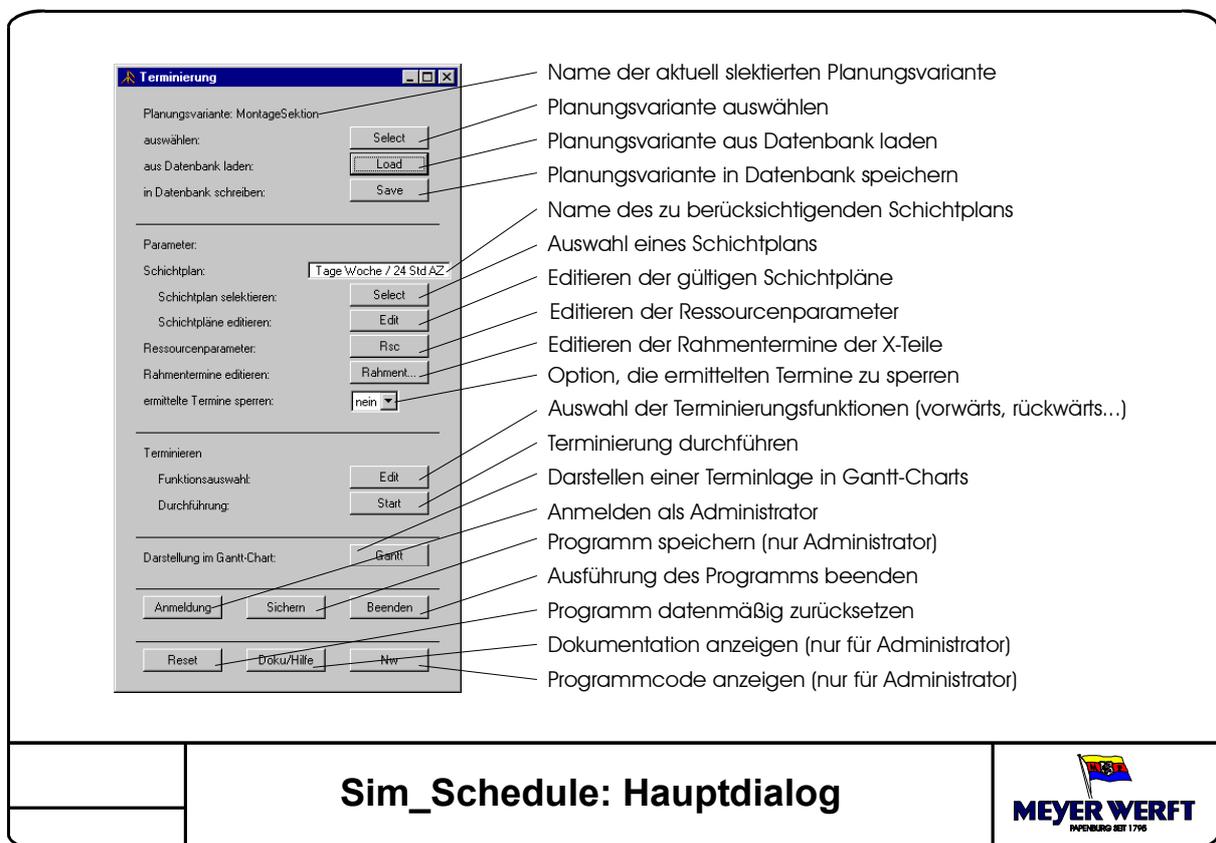


Abbildung 81 - Sim_Schedule: Hauptdialog

2.4.1.1.1 Vorbereiten der Terminierung

Variante auswählen und laden

Zunächst wird eine Planungsvariante ausgewählt und aus der Datenbank eingelesen („Select“- und „Load“-Button des Hauptdialogs). Der Name der Variante wird im Dialog angezeigt.

Parameter der Terminierung setzen

Eine Reihe von Randbedingungen des Terminierungsablaufs kann nun festgelegt werden:

- Schichtkalender: Vor der Terminierung wird ein global geltender Schichtkalender festgelegt, der grundsätzlich auf alle Arbeitsschritte anzuwenden ist. Abweichend können aber auch jeder Ressource spezielle Schichtkalender zugeordnet werden.
- Rahmentermine: Den Arbeitsschritten der höchsten Elemente der Produktstruktur (Schiffe, Bauabschnitte) werden Rahmentermine (Meilensteine) zugeordnet. Der in Abbildung 82 gezeigte Dialog bietet die Möglichkeit, diese Rahmentermine zu überprüfen bzw. mit Hilfe des ActiveX-Elements DatePicker (2.2.8) zu ändern.
- Auszuführende Funktionen: Die Terminierung bietet eine Reihe gekapselter Funktionen, die u.U. nicht alle ausgeführt werden sollen. Hier lässt sich eine Auswahl der auszuführenden Funktionen treffen (Button „Funktionsauswahl Edit“, Abbildung 83).
- Ressourcenparameter editieren: Relevant für die Vorwärtsterminierung, wenn der Takt bestimmter Ressourcen als Maßgabe für den Frühesten Start des ersten Arbeitsschritts von Bauteilen gelten soll.
- Termine sperren: Es kann ausgewählt werden, ob die ermittelten Termine in die Datenbank mit dem Vermerk „gesperrt“ zurückgeschrieben werden sollen.

Zu anderen Leveln der Produktstruktur wechseln
(bei X-Teilen in mehr als drei Leveln)

Rahmentermine

Planungsvariante: MontageSektion

höheren Level anzeigen: < | > | >>

tieferen Level anzeigen: << | < | >

Level	MontageSektion_656	Spätester Start:	21.03.2001 00:00:00.0000
Level1	Typ: Kreuzfahrer	Spätestes Ende:	03.12.2001 00:00:00.0000
	ID_Bauteil: MontageSektion_656	Dauer:	160:00:00.00000
	Arbeitsschritt: MontageSektion_Schiffsmontage-1	Anzahl taktbestimmende Teile:	50
	<input type="button" value="Edit"/>	Takt:	3:04:48:00.0000
Level2	MontageSektion_656_2	Spätester Start:	31.05.2001 00:00:00.0000
	Typ: Bauabschnitt	Spätestes Ende:	23.11.2001 00:00:00.0000
	ID_Bauteil: MontageSektion_656_2	Dauer:	109:00:00.00.0000
	Arbeitsschritt: MontageSektion_Bauabschnittbau	Anzahl taktbestimmende Teile:	48
	<input type="button" value="Edit"/>	Takt:	2:06:30:00.0000
Level3	MontageSektion_656_2_1	Spätester Start:	21.03.2001 00:00:00.0000
	Typ: Bauabschnitt	Spätestes Ende:	31.05.2001 00:00:00.0000
	ID_Bauteil: MontageSektion_656_2_1	Dauer:	45:00:00.00.0000
	Arbeitsschritt: MontageSektion_Bauabschnittbau	Anzahl taktbestimmende Teile:	11
	<input type="button" value="Edit"/>	Takt:	4:02:10:54.5455

Date Picker

Bauteil: MontageSektion_656_2_1
Arbeitsschritt: MontageSektion_Bauabschnittbau

Start: 21.03.01

Ende: 31.05.01

Mo Di Mi Do Fr Sa So

30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

Heute: 02.12.02

Name der aktuell geladenen Planungsvariante

Informationen zum ausgewählten Bauteil (Level in Produktstruktur, ID des Bauteils, Typ)

Auswahl eines Bauteils

Auswahl eines Arbeitsschritts

Termine des Arbeitsschritts bearbeiten, Verwendung des Datepickers (s.o.)

Rahmentermine und Dauer des ausgewählten Arbeitsschritts

falls dem ausgewählten Arbeitsschritt Montageschritte zugeordnet sind: Anzahl untergeordneter Y-Teile und resultierender Takt

Sim_Schedule: Dialog Rahmentermine



Abbildung 82 - Sim_Schedule: Dialog Rahmentermine



Abbildung 83 - Sim_Schedule: Festlegen der auszuführenden Funktionen

2.4.1.1.2 Terminieren der Baufolge

Die originäre Funktion des Modells ist das Terminieren der Fertigungsschritte, die den (stahlbaulichen) Zusammenbau von Schiffen abbilden. Es handelt sich um lokale Netzpläne, die jeweils auf ein Element (Bauteil, Baugruppe, KTR) bezogen sind. Spezielle Montage-Arbeitsschritte mit definierten Eingangsbauteilen bilden die Bindeglieder zwischen den lokalen Plänen.

2.4.1.1.3 Rückwärtsterminierung

Taktung / Abgleich

Dieser erste Schritt im Rahmen der Rückwärtsterminierung wertet die vorgegebenen Rahmentermine aus und ermittelt für jedes nächst untergeordnete Teil der Variante den letzten Arbeitsschritt und zu diesem den Spätesten Termin.

Jeweils zwei Rahmentermine eines Elements geben ein Zeitintervall vor, in dem die Montage aus den untergeordneten Bauteilen erfolgen soll. Da die Anzahl der einzubauenden Teile bekannt ist, ergibt sich ein Einbautakt. Sofern die Funktion „Abgleich“ deaktiviert ist, werden die Termine für die Montageschritte der Unterbauten entsprechend der Montagereihenfolge jeweils um den Takt versetzt errechnet.

Ist die Abgleichfunktion dagegen aktiviert, so werden zunächst die Rahmentermine der Planungsvariante miteinander verglichen und Phasen gegenseitiger zeitlicher Überlappung des Zusammenbaus ermittelt. Ohne Abgleich gäbe es - bezogen auf die Werft insgesamt - eine Folge von Phasen, in denen ein, zwei oder noch mehr Elemente mit Unterbauten „beliefert“ werden müssten. Die entsprechenden Fertigungsressourcen wären stark schwankenden Belastungen ausgesetzt. Die Abgleichsfunktion variiert in einem iterativen Verfahren je Kostenträger bzw. Bauabschnitt die Takte in den einzelnen Phasen, um eine für den Betrieb bessere Situation zu schaffen. Dies ist keine Optimierung, aber eine geeignete Maßnahme, um für spätere Optimierungen eine verbesserte Startlösung zu finden. In Abbildung 84 ist das Prinzip des Abgleichens dargestellt.

Die Ergebnisse werden in den Datenbaustein zurückgeschrieben.

Durchlaufterminierung

Nachdem die Spätesten Termine für die Fertigstellung der Y-Teile bekannt sind, werden in der Durchlaufterminierung rekursiv alle noch fehlenden Spätesten Termine ermittelt. Rekursiv werden dabei über die Montageschritte und Eingangsbauteile alle Stufen der Hierarchy of

Assembly durchschritten. Anschließend liegen alle Spätesten Termine der auf die Sicht „Bau-
folge“ bezogenen Arbeitsschritte im Datenbaustein vor.

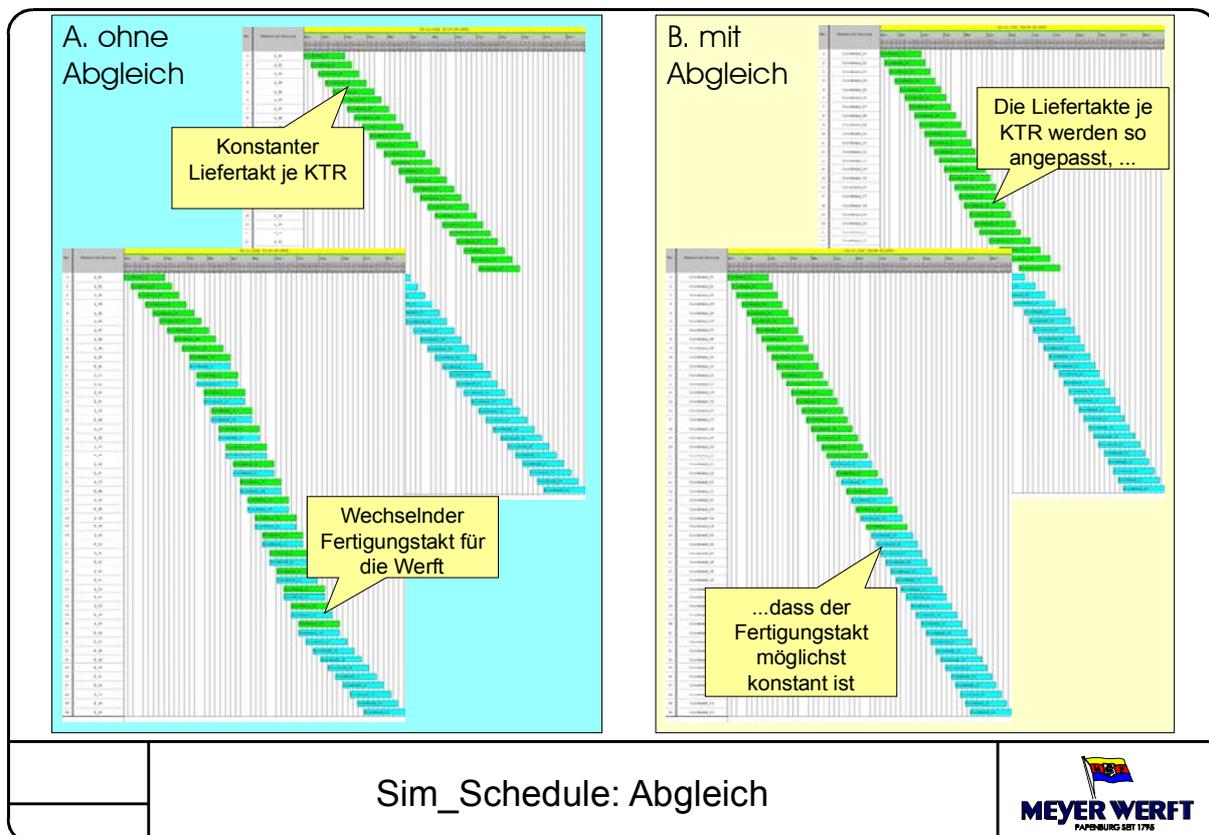


Abbildung 84 - Sim_Schedule: Abgleich

2.4.1.1.4 Vorwärtsterminierung

Terminanbindung

Die Terminanbindung im Rahmen der Vorwärtsterminierung ermittelt für jedes Teil der Variante den ersten Arbeitsschritt und zu diesem den Frühesten Termin. Zur Findung dieses Termins können unterschiedliche Methoden zum Einsatz kommen.

- festes Zeitfenster:
Dieses einfache Verfahren zielt darauf ab, allen betrachteten Arbeitsschritten ein bestimmtes Zeitfenster einzuräumen. Dazu wird der bereits bekannte Späteste Starttermin aus der Datenstruktur ausgelesen und - unter Berücksichtigung des Schichtkalenders und eines definierbaren Vorlaufs - der Früheste Starttermin errechnet.
- getaktete Ressourcen:
In der Regel ist in der Datenstruktur dem jeweils ersten Arbeitsschritt eines Bauteils eine Ressource als Arbeitsort bereits eindeutig zugeordnet. Die Datenstruktur erlaubt es, jeder Ressource einen zeitlich veränderlichen Arbeitstakt zuzuweisen. Es wurde ein Algorithmus implementiert, der
 - prüft, welche Bauteile auf der betrachteten Ressource bearbeitet werden sollen und bereits mit gesperrten Terminen versehen sind,
 - welche Termine aufgrund dieser Vorbelegung und des variablen Takts der Ressource noch belegbar sind,

- welche Bauteile, die auf dieser Ressource zu bearbeiten sind, noch terminiert werden müssen und
- welche Bearbeitungsreihenfolge aufgrund der bereits bekannten Spätesten Starttermine sinnvoll ist.

Die freien Termine werden entsprechend der so gefundenen Reihenfolge auf die Bauteile verteilt.

Durchlaufterminierung

Nachdem die Frühesten Termine für die Erzeugung der Teile bekannt sind, werden in der Durchlaufterminierung rekursiv alle noch fehlenden Frühesten Termine aller Arbeitsschritte ermittelt.

- Es wird eine Tabelle mit allen Bauteilen eines Kostenträgers angelegt. Die Arbeitsschritte und ihre Anordnungsbeziehungen werden in Untertabellen geschrieben.
- Zu jedem Bauteil wird der erste Arbeitsschritt bestimmt. Mit dem Frühesten Start dieses Schritts wird die Vorwärtsterminierung aller Arbeitsschritte des Bauteils begonnen.
- Abschließend werden die Frühesten Termine in die Tabelle „Ap“ geschrieben.

2.4.1.1.5 Terminieren hierarchischer Netzpläne

Die oben beschriebenen Algorithmen betreffen die Terminierung lokaler Netzpläne, die jeweils auf ein bestimmtes Element der Hierarchy of Assembly bezogen sind. Die lokalen Netzpläne selbst sind nicht hierarchisiert. Die Datenstruktur sieht aber auch die Möglichkeit vor, hierarchisch organisierte Netzpläne zu modellieren. Zudem kann jeder einzelne Arbeitsschritt auf ein beliebiges Element der Produktstruktur (Topologie und Sicht) bezogen sein.

In einem rekursiven Verfahren werden die hierarchisch aufgebauten Netzpläne terminiert. Dazu werden nacheinander alle Arbeitsschritte betrachtet, die der Hierarchie 1 angehören. Es wird geprüft, ob dem jeweiligen Arbeitsschritt weitere Arbeitsschritte hierarchisch direkt untergeordnet sind. Ist dies der Fall, so wird der untergeordnete Netzplan terminiert. Dabei ist der Früheste Start des übergeordneten Arbeitsschritts der Bezugstermin für die Vorwärtsterminierung, das Späteste Ende des für die Rückwärtsterminierung. Nachdem die Termine der untergeordneten Arbeitsschritte fest stehen, wird rekursiv nach weiteren untergeordneten Netzplänen gesucht, um diese ebenfalls zu terminieren.

2.4.1.2 Sim_Space (Raumbelegung)

Wie bereits erläutert, wurde für die Belegungsplanung der Meyer Werft der Raumbaustein entwickelt. Im Modell „Sim_Space“ wurde mit Hilfe des Raumbausteins ein Modell der Meyer Werft entwickelt und dem Anwender in der Pilotphase zur Verfügung gestellt. Mit dem Simulationsmodell sollen folgende Aufgaben und Fragestellungen bearbeitet bzw. untersucht werden:

- Projektübergreifende Ressourcenplanung
- Berücksichtigung von Regeln zur Bearbeitung der Aufträge
- Durchspielen von Varianten (Experimentierfähige Simulationsumgebung)
- Terminverschiebungen aufgrund von Kapazitätsengpässen
- Überprüfung von Vergabestrategien

Aufbau des Modells

Das Modell wurde ausschließlich mit Bausteinen aus dem Simulationstoolset entwickelt. Folgende Bausteine wurden im Modell verwendet:

- Raumbaustein
Der Raumbaustein wurde zum Abbilden der verschiedenen Ressourcen (Hallen) verwendet. Im Modell wurde die gesamte Werft mit den Hallen II – VI modelliert.
- Lifecycle Baustein
Der Lifecycle Baustein wurde verwendet, um die in der Simulation erzeugten Raumbelegungsdaten in einer Belegungsvariante im Datenmodell (Datenbank) zu speichern.
- Pausenbaustein
Wie in allen Modellen, die im Projekt entwickelt wurden, wurde der Pausenbaustein für die zeitlichen Ablaufsteuerungen (Arbeitszeiten, Urlaubszeiten etc.) verwendet.
- Datenbausteine
Für die Anbindung an das Datenmodell werden die Datenbausteine verwendet.
- Leitreechnerbaustein
Die Steuerung im Modell konnte, wie im Konzept bereits beschrieben, von einer Leitreechnerkomponente übernommen werden.

Das Modell „Sim_Space“ konnte vollständig auf der Basis des Simulationstoolsets erstellt werden. Eine Kodierung im Modell war bei der Erstellung des Modells nicht erforderlich. Das gesamte Modell wurde im ersten Schritt interaktiv (Drag and Drop) mit den Simulationsbausteinen aufgebaut. Im zweiten Schritt wurden dann über die entsprechenden Dialoge die verwendeten Bausteine parametrisiert.

Dem Anwender wird beim Start des Modells die entsprechende Modellsteuerung bereitgestellt. Mit der Modellsteuerung kann das Modell vollständig bedient werden. Der Einsatz erfolgt mit einer Laufzeitlizenz, da der Anwender nicht auf der Expertenebene ins Modell eingreifen muss.

Mit diesem Modell konnte sehr deutlich aufgezeigt werden, dass eine Modellerstellung auf der Basis des Simulationstoolset ohne Probleme möglich ist. Die Modellierungszeit für ein sehr komplexes Werftmodell konnte somit durch die Verwendung des Simulationstoolsets drastisch reduziert werden. Änderungen, die sich in der Pilotphase ergaben, konnten zusammen mit den Anwen- den in sehr kurzer Zeit umgesetzt werden. Abbildung 85 zeigt das Simulationsmodell „Sim_Space“.

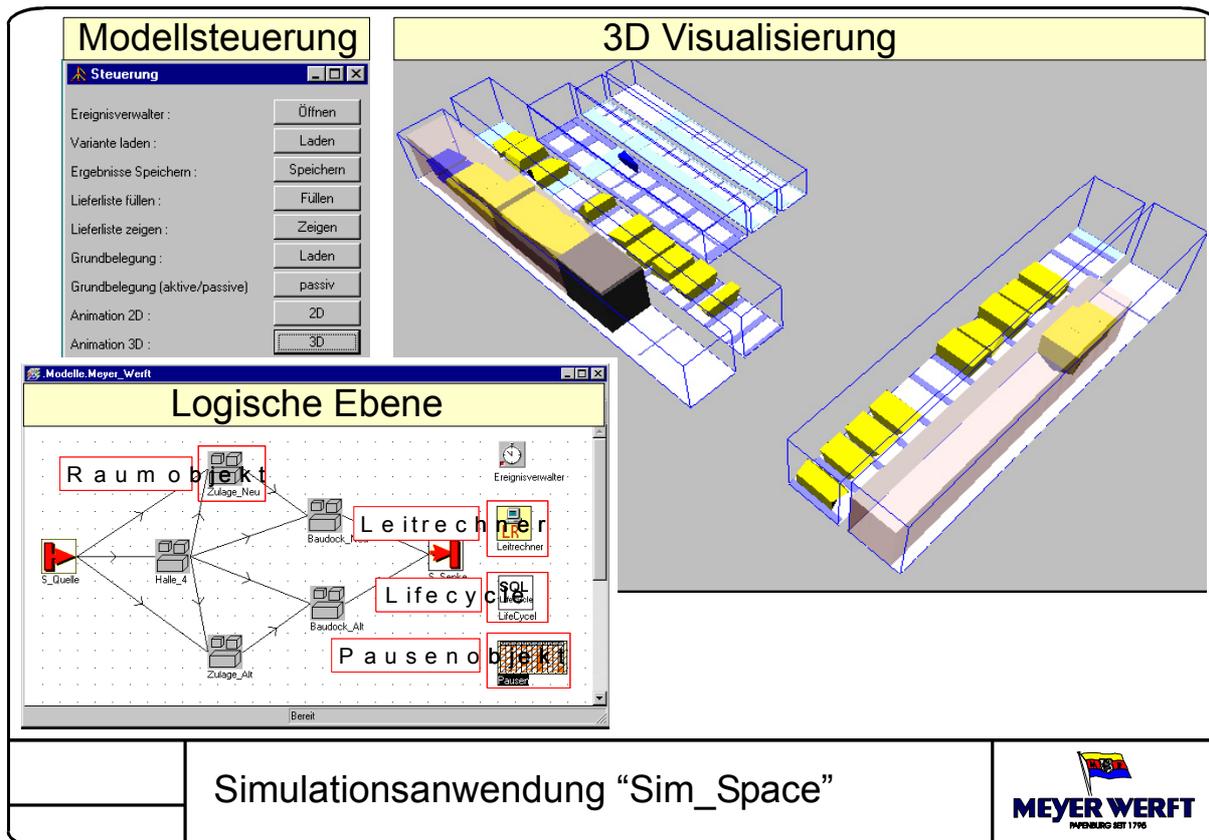


Abbildung 85 – Sim-Space: Übersicht

Entwicklungsstufen

Die Simulationsanwendung „Sim_Space“ war während der gesamten Projektlaufzeit die Pilotanwendung für das gesamte Toolset. Neue Konzepte und Entwicklungen wurden oftmals im Umfeld dieser Anwendung getestet und realisiert. Das Modell wurde im Projekt zuerst auf Expertenebene in der Fabrikplanung (Investitionsplanung) eingesetzt. In der Pilotphase wurde das Modell mit der entsprechenden Umgebung Anwendern zur Verfügung gestellt. Mit diesem Einsatz des Toolsets in den Bereichen Investitionsplanung und Fertigungsplanung gleichermaßen konnte die Richtigkeit der im Projekt verfolgten Strategie aufgezeigt werden.

Erzielte Ergebnisse

Mit dem Simulationsmodell kann die Flächenbelegungsplanung der gesamten Werft unterstützt werden. Folgende Ergebnisse und Verbesserungen konnten mit dem Einsatz der Simulationslösung erzielt werden:

- Verringerung der Planungszeiten
- Gute Entscheidungsgrundlage für Vergabestrategien
- Verbesserte Ressourcenausnutzung
- Projektübergreifende Ressourcenplanung
- Erhöhe Planungssicherheit.

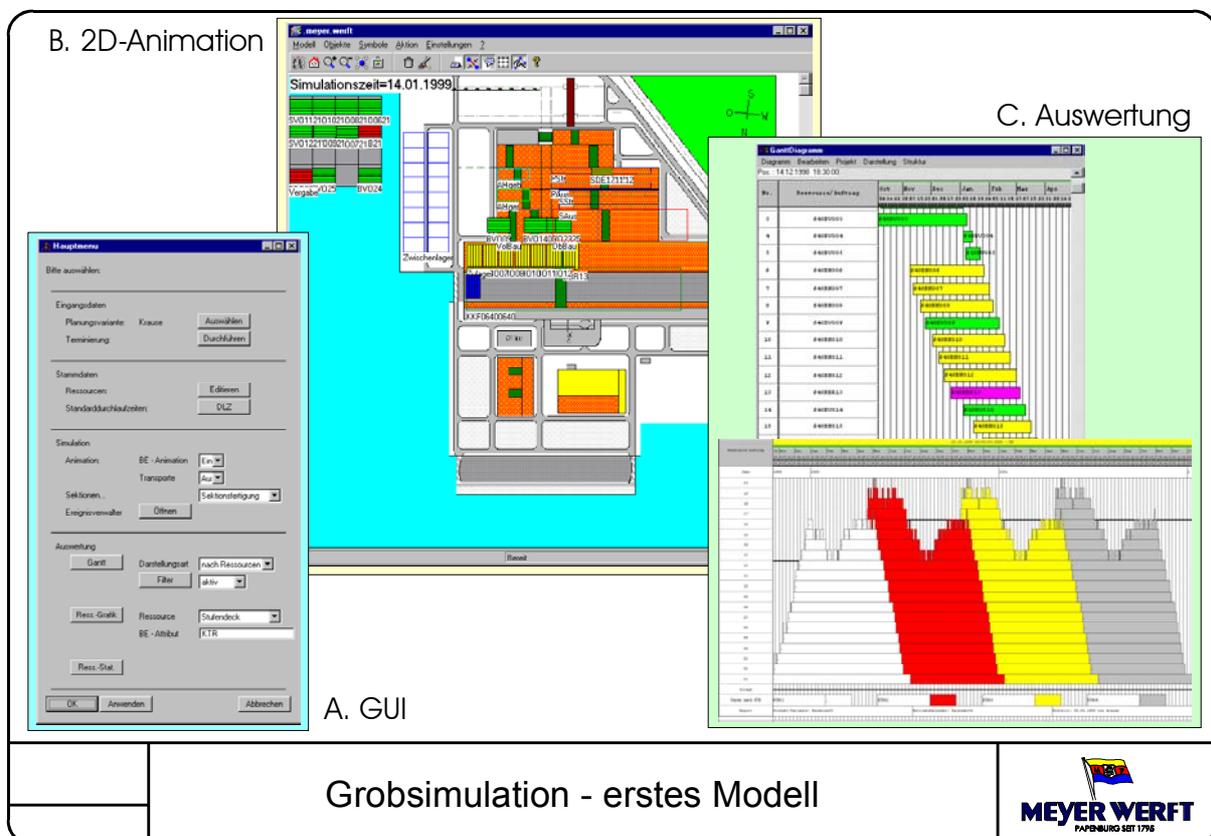
2.4.2 Fabrikplanung (Expertenebene)

Für den klassischen Simulationsanwendungsfall im Bereich der Fabrikplanung wurden im Laufe des Projektes einige Modelle erstellt. Basis für diese Modelle war das entwickelte Simulationstoolset. Die Simulationsstudien und Simulationsexperimente wurden vom Simulationsexperten in Abstimmung mit der Fachabteilung durchgeführt.

2.4.2.1 Grobsimulation

Für die Meyer Werft als Kompaktwerft ist die Fläche eine Engpassressource. Eine häufig wiederkehrende Planungsaufgabe ist daher die überschlägige Ermittlung des Bedarfs an Block- und Sektionsbauplätzen sowie der Baudockbelegung über einen längeren Zeitraum (Monate bis Jahre). Diese Tätigkeit wird sowohl im Hinblick auf die langfristige Planung des Bauprogramms als auch zur Prüfung von Investitionsvorhaben (Werftplanung) durchgeführt. Dabei ist es oft erforderlich, zu einem Planungsszenario in kurzer Zeit mehrere nur geringfügig variierende Untervarianten zu überprüfen.

Zu Beginn des Projekts wurde diese „Grobplanung“ üblicherweise mit diverser Standardsoftware (Corel Draw, Excel, MS-Project) oder einfach „mit Papier und Bleistift“ betrieben. Lediglich einzelne Arbeitsgänge waren in Form von Programmen teilautomatisiert. Die Simulation wurde in dieser Zeit als eine Technik erkannt, die geeignet ist, in kürzerer Zeit zu verlässlichen Ergebnissen zu gelangen. Auch mit der Absicht, erste Erfahrungen mit der Basissoftware eM-Plant zu sammeln und erste Komponenten für das Toolset zu erstellen, wurde die Entwicklung eines Grobsimulationsmodells begonnen.



Grobsimulation - erstes Modell

Abbildung 86 - Grobsimulation: erstes Modell

Realisierung

Teile des ersten Modells zur Grobsimulation sind in Abbildung 86 dargestellt. Da in kurzer Zeit ein Simulationsmodell benötigt wurde und noch erste Erfahrungen im Umgang mit eM-Plant gesammelt werden mussten, wurde hauptsächlich auf herkömmliche Weise mit den vorhandenen Basisbausteinen modelliert. Allgemeine und wiederverwendbare Bausteine wurden in sehr geringem Maße erstellt. Das Modell lässt sich kurz wie folgt charakterisieren:

- Datenbankanbindung: SQL-Schnittstelle zum Einlesen von Planungsvarianten (Datenmodell)
- Ressourcen: Platzlogik für Block- und Sektionsbau; Baudocks nur zu Visualisierungszwecken (keine begrenzte Kapazität)
- Arbeitspläne: Die Arbeitspläne der Bauteile wurden durch Typarbeitspläne bestimmt. Um spezielle Logiken wie z.B. bei Überbelegung von Ressourcen zu modellieren, wurden zahlreiche zusätzliche Methoden programmiert.
- Auswertung: Hier wurden erste allgemeine Bausteine eingesetzt, die für das Simulationstoolset entwickelt wurden (2.2.7.2)

Das Grobsimulationsmodell wurde in der ersten Phase des Projektes im Bereich der Fabrikplanung eingesetzt. Mit dem Modell konnten in zahlreichen Simulationsstudien sehr gute Ergebnisse erzielt werden. Als Eingangsdaten standen zum damaligen Zeitpunkt nur die Produktdaten zu Verfügung. Arbeitspläne und Ressourcendaten wurden noch in Modell vom Simulationsexperten hinterlegt. Abbildung 86-C zeigt zwei typische Auswertungen (Gantt- und Kapazitätsdiagramm), die mit dem Modell erstellt wurden.

Aus dem Modell konnten viele Bausteine wie z.B. der Pausenbaustein und der Ganttbaustein für das Simulationstoolset abgeleitet werden.

Die Entwicklung des Grobsimulationsmodells war jedoch nur ein Zwischenschritt. Ziel war es, im Gegensatz zum platzorientierten Grobsimulationsmodell ein flächenorientiertes Modell zu entwickeln. Dies wurde im Projekt mit dem Modell „Sim_Space“ (2.4.1.2) realisiert.

2.4.2.2 Paneelstraße

Bei der Vorbereitung von Investitionsentscheidungen hat die Projektgruppe Simulation die Werftplanung im Vorfeld des Aufbaus einer neuen Paneelstraße aktiv unterstützt. Ein Simulationsmodell des Fertigungsbereichs konnte zum Teil aus bereits erstellten Bausteinen aufgebaut werden, zum Teil waren Anpassungen oder Neuentwicklungen erforderlich. Die Simulationsstudie diente der Erhöhung der Planungssicherheit in der Phase der Anlagenspezifikation.

Prozesse

Das Modell bildet fünf Stationen der Paneelstraße ab (Abbildung 87):

1. Station "Besäumen": Mehrere Bearbeitungsprozesse, die an einzelnen Stahlplatten vorgenommen werden
2. Station "Einseitenschweißen quer": Verschweißen von jeweils zwei Platten zu einem Blechstreifen (Nähte in Querrichtung des Schiffes) unter einem feststehenden Portal
3. Station "Einseitenschweißen längs": Verschweißen mehrerer Streifen zum Plattenplan (Nähte in Längsrichtung des Schiffes) unter einem feststehenden Portal

4. Station "Plattenplan bearbeiten": Mehrere Bearbeitungsprozesse, die von einer beweglichen Arbeitsmaschine am Plattenplan durchgeführt werden
5. Station "Profile schweißen": Aufschweißen von Hollandprofilen unter einem feststehenden Portal

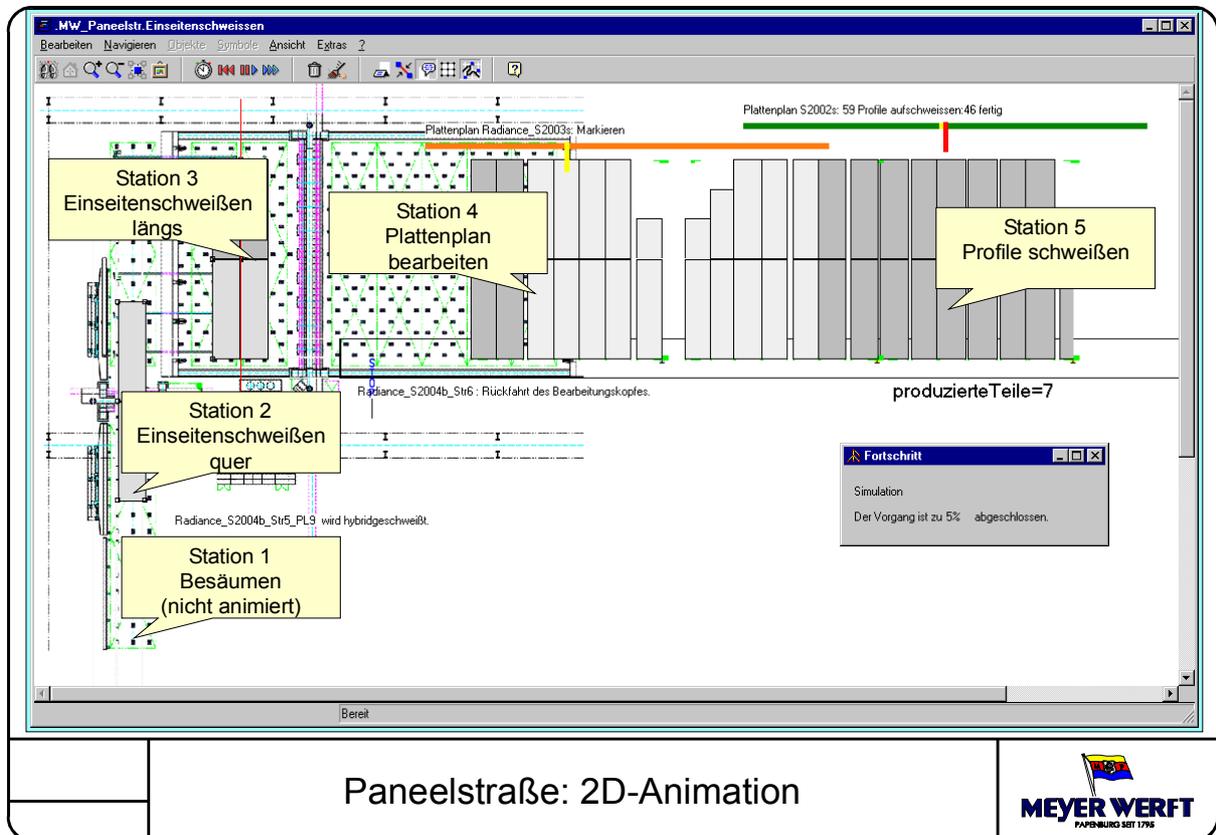


Abbildung 87 - Simulation Panelstraße: 2D-Animation

Modellierung

Die Stationen 1 bis 3 wurden mit eigens für dieses Modell neu entwickelten Bausteinen abgebildet, bei der Modellierung der Stationen 4 und 5 wurden die Elemente des in einer Diplomarbeit entwickelten Bausteinkastens "Panelstraße" verwendet.

Ressourcen- und Prozessdaten

Auf jeder der genannten Stationen wird eine Anzahl sequentiell auszuführender Prozesse durchlaufen. Je Station wurden Tabellen angelegt, in denen die Prozessparameter (konstante Zeiten, Gleichungen, Kennlinien) geführt und vom GUI aus editiert werden können (Abbildung 88 - 6).

Trotz des hohen Automatisierungsgrades der Anlagen ist die Überwachung der Prozesse durch das Bedienpersonal erforderlich, und einige Handhabungstätigkeiten müssen weiterhin manuell ausgeführt werden. Daher wurde je Prozess die benötigte relative Einbindung der Ressource „Bediener“ berücksichtigt (z.B. 20% \approx gelegentliche Überwachung des Prozesses, 100% \approx Bediener muss den Prozess manuell durchführen bzw. die ganze Zeit überwachen).

Die Spezifikation der einzelnen Stationen war zur Zeit der Simulationsstudie unterschiedlich weit fortgeschritten, gleiches galt für die Kenntnis der einzelnen Prozesse. Mit einem Ausgleichsfaktor (hier als „Einschaltdauer“ ESD bezeichnet) wurden Unsicherheiten ausgegli-

chen. Stationen, auf denen z.B. bereits einzelne Handhabungs-, Positionier- und Wartungsvorgänge bekannt und im Simulationsmodell modelliert waren, wurden mit einer ESD nahe 100% bewertet, während Stationen, bei denen lediglich die Parameter des eigentlichen Bearbeitungsprozesses bekannt waren, mit einer kleineren ESD berücksichtigt wurden.

Die ESD der Stationen und die Anzahl verfügbarer Bediener kann im GUI festgelegt werden (Abbildung 88 - 5).

Produktdaten

Für das Modell mussten simulationsgerechte Stücklisten erzeugt werden. Die Erzeugung der Stücklisten kann auf zwei Wegen erfolgen. Zum einen wurden anhand von Sektionszeichnungen gebauter Schiffe die statistischen Verteilungen der interessierenden Merkmale ermittelt, wie z.B. Hauptabmessungen der Sektionen und der einzelnen Platten, Häufigkeit von Blechdickensprüngen in Quer- und Längsnähten, Geometrietyp des Plattenplans (rechteckig, L- oder O-förmig etc.) sowie relative Lage von Ausschnitten. Es wurde ein eM-Plant- Baustein erstellt, mit dem anhand dieser Verteilungen Stücklisten "künstlich" erstellt werden konnten (Abbildung 88 - 3). Die Inhalte der so erhaltenen Stücklisten sind zwar nicht Daten realer Schiffe, sie erfüllen jedoch mit guter Näherung die in der Realität vorhandenen statistischen Verteilungen.

- **Vorteil:**

Planungsvarianten mit beliebig vielen Sektionen können in kurzer Zeit erstellt werden, nachdem einmal die Verteilungen ermittelt worden sind.

- **Nachteile:**

Obwohl das Ergebnis der Stücklistengenerierung in bezug auf die statistische Verteilung aller erfassten Parameter korrekt ist, werden vereinzelt "unrealistische" Sektionen erzeugt. Dies liegt daran, dass die Abhängigkeiten der untersuchten Parameter voneinander nicht vollständig erfasst werden konnten. Eine manuelle Kontrolle der Stücklisten wird so erforderlich.

Das beschriebene Verfahren liefert einen Produktmix, in dem arbeitsintensive und weniger komplizierte Bauteile mehr oder weniger gleichmäßig verteilt sind. Die zeitweise Anhäufung der aufwendigen Bauteile, zu der die Fertigungsfolge u.U. zwingt, wird hier außer acht gelassen.

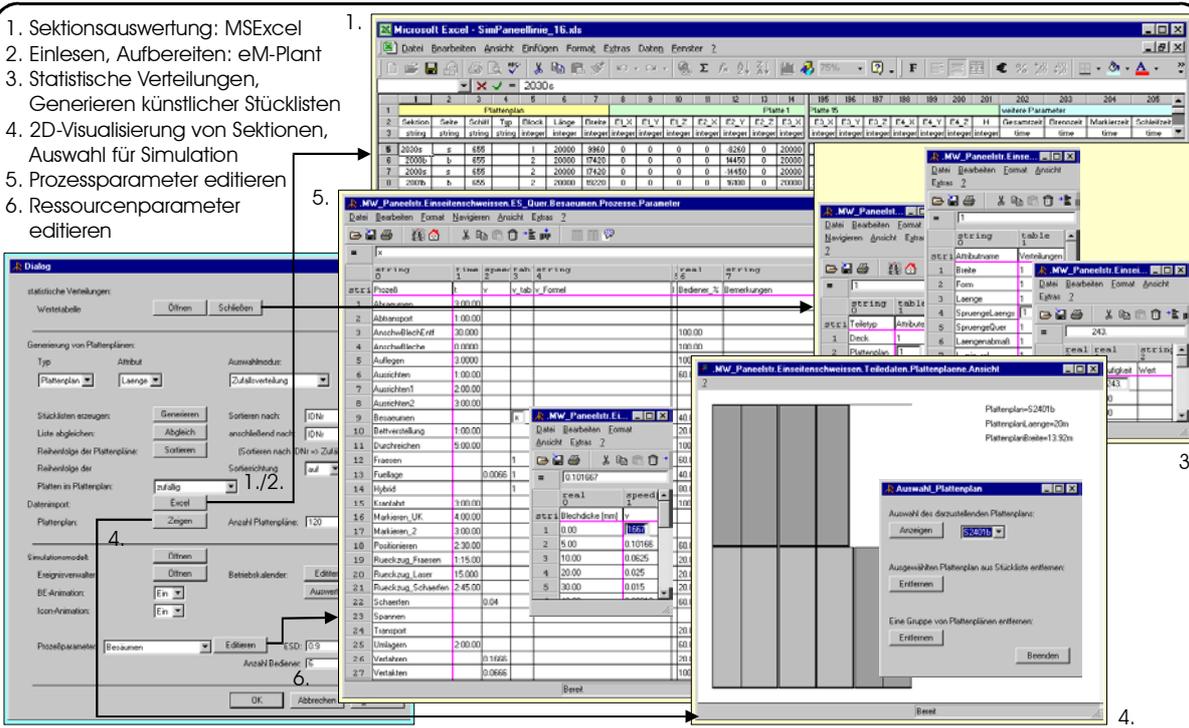
Um auch die Fertigung realer Decksektionen simulieren zu können, wurden zusätzlich Zeichnungen eines z.Z. im Bau befindlichen Kostenträgers in MSExcel manuell ausgewertet und der Simulation zur Verfügung gestellt (Abbildung 88 - 1,2).

- **Vorteile:**

Die Simulation der Fertigung realer Bauteile führt zu höherer Akzeptanz gegenüber der Simulation, und das Verhalten der Panelstraße bei realistischen Produktionsreihenfolgen kann untersucht werden.

- **Nachteile:**

Das Erstellen neuer Planungsvarianten bedeutet jedesmal einen erheblichen Aufwand, da Bauteile von Hand ausgemessen werden müssen.



The screenshot displays the eM-Plant software interface with several windows open. A central Excel spreadsheet window shows a table with columns for 'Plattenplan' and 'Länge'. Other windows include 'MW_Panelstr.Einzelherschweissen.E5_Daten Ressourcen Prozesse Parameter', 'MW_Panelstr.Einzelherschweissen.Teilendaten.Plattenpläne Ansicht', and 'Auswahl_Plattenplan'. A 'Dialog' window is also visible on the left side.

1. Sektionsauswertung: MSEXcel
2. Einlesen, Aufbereiten: eM-Plant
3. Statistische Verteilungen, Generieren künstlicher Stücklisten
4. 2D-Visualisierung von Sektionen, Auswahl für Simulation
5. Prozessparameter editieren
6. Ressourcenparameter editieren

Panelstraße: Eingangsdaten, Parametrierung



Abbildung 88 - Simulation Panelstraße: Eingangsdaten, Parametrierung

Beide Verfahren wurden bei der Generierung der simulationsgerechten Daten angewendet.

Visualisierung

Zwecks anschaulicher Visualisierung des Zusammenbaus von Plattenplänen wurde ein Baustein programmiert, der zur Simulationslaufzeit bei jedem Füge- bzw. Montagevorgang die Animationsstruktur des neu entstehenden Bauteils entsprechend der Geometriedaten der beteiligten Eingangsbauteile setzt. Durch unterschiedliche Graustufen werden verschiedene Blechdicken sichtbar gemacht, so dass bereits im 2D-Modus wichtige Informationen zu den Plattenplänen sichtbar gemacht werden können. Im eM-Plant-3D-Viewer kann auch der weitere Zusammenbau zum Panel mit Versteifungen sichtbar gemacht werden.

Zusätzlich besteht die Möglichkeit, Plattenpläne im Vorfeld der Simulation in 2D zu visualisieren (Abbildung 88 - 4). So können dann einzelne Sektionen aus der Stückliste entfernt werden, z.B. wenn bei der Generierung künstlicher Stücklisten besonders unwahrscheinliche Exemplare erstellt wurden. Zur besseren Transparenz der simulierten Fertigungsschritte wurde noch ein Baustein erstellt, der zur Simulationslaufzeit die aktuellen Prozesse und die Namen der beteiligten Bauteile in Kommunikationsfeldern anzeigt.

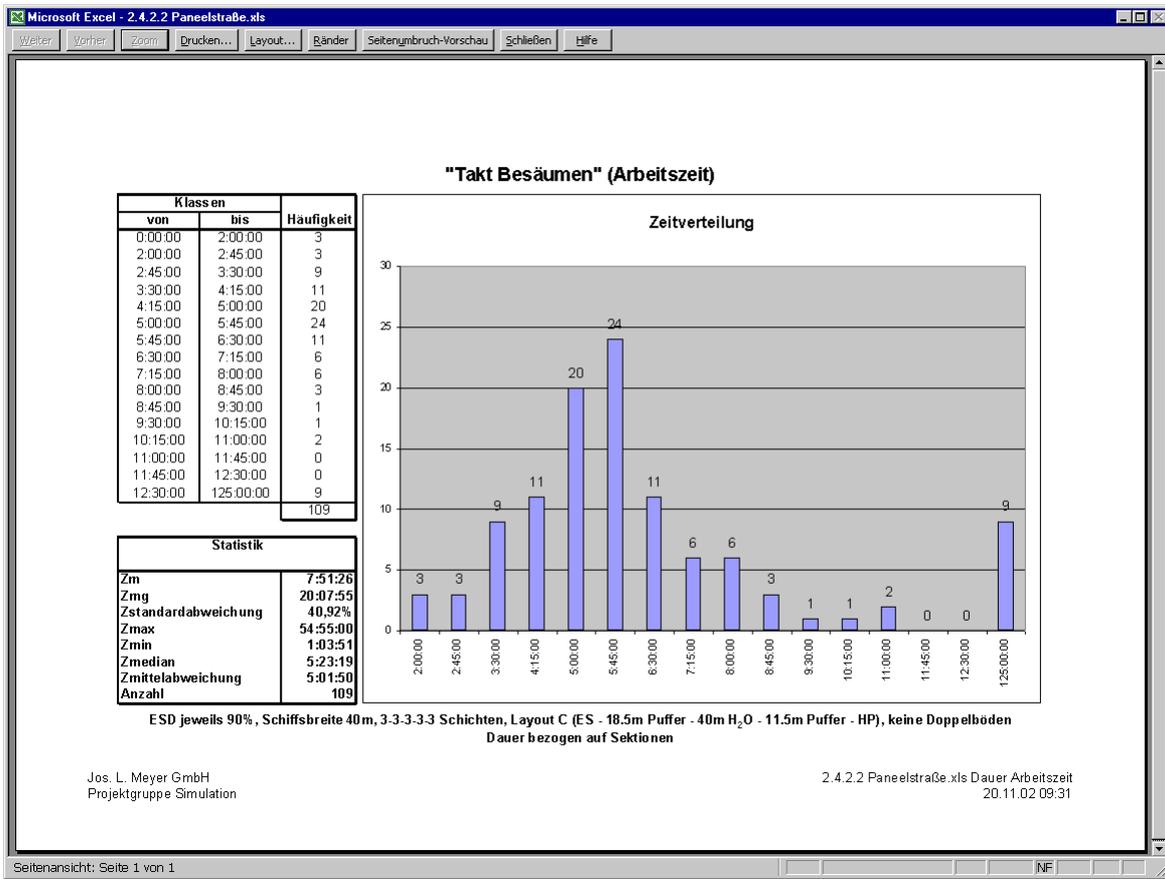


Abbildung 89 - Simulation Panelstraße: Auswertung

Experimente

Mit dem Simulationsmodell wurden folgende Fragestellungen untersucht:

- Erreichbarkeit der geforderten Takt- und Durchlaufzeiten der Stationen, Ableiten von Anforderungen für einzelne Prozesszeiten und -geschwindigkeiten
- Ermittlung der Anzahl benötigter Maschinenbediener auf den einzelnen Stationen
- Einfluss unterschiedlicher Schichtzeiten auf den verschiedenen Stationen mit Hilfe des Pausenbausteins
- Ermittlung geeigneter Werte für die Hauptabmessungen durch Variation von Arbeitsbereichen und Puffern.
- Durchlaufverhalten bei unterschiedlichen Produktmischen
- Abschätzung des Bedarfs an Bedienpersonal

2.5 Unterstützende Verfahren

2.5.1 CAD Datentransfer

Für die frühzeitige Belegungsplanung kann unter vertretbarem Aufwand ein neues Schiffbauprojekt auf der Ebene von Blöcken und Sektionen noch manuell erzeugt werden. Die manuelle Erzeugung der Daten wird vom SDM Tool unterstützt, so dass für ein neues Schiffbauprojekt mit ca. 65 Blöcken und ca. 350 Sektionen ein Aufwand von maximal zwei Tagen entsteht. Diese kann komplett ohne eine CAD-Anwendung erfolgen.

2.5.1.1 Anforderungen

Die Modellierung von Produktdaten auf der Ebene von Einzelteilen kann nicht manuell erfolgen, da die Anzahl der Elemente und die Hierarchiestufen auf Einzelteilebene explosionsartig zunehmen. Eine Sektion wird durch ca. 5 Hierarchiestufen abgebildet und besteht aus ca. 700-1000 Einzelteilen. Für den CAD-Datentransfer wurde im Projekt eine Schnittstelle zum CAD-System CATIA von Dassault Systems aufgebaut. Folgende Anforderungen wurden an die Schnittstelle gestellt:

- **Acht-Punkt-Geometriemodell**
Das vollständige Acht-Punkt-Geometriemodell soll für alle Einzelteile erzeugt werden.
- **Hierarchy of Assembly**
Die Stückliste muss als hierarchische Stückliste abgebildet werden, damit Zwischenprodukte die in der Fertigung entstehen auch in der Simulation berücksichtigt werden können.
- **Lage der Einzelteile im Schiff**
Alle Einzelteile sollten mit ihren Koordinaten auf die absolute Lage im Schiff (Nullspant) bezogen werden.
- **Zusätzliche Attribute**
Zusätzliche Attribute wie z.B. Länge, Breite, Höhe oder Bauteiltyp etc. sollen von der Schnittstelle übertragen werden.
- **Bereitstellung von VRML-Dateien**
Für eine 3D-Visualisierung ist eine Darstellung der einzelnen Elemente mit Hilfe von VRML-Dateien geplant. Die Dateien müssen vom CAD-System bereitgestellt werden.

2.5.1.2 Realisierung

Ziel dieser Entwicklung war nicht die Entwicklung einer allgemeingültigen CAD-Schnittstelle (vgl. STEP) sondern die Entwicklung einer CAD-Schnittstelle, die auf die Bedürfnisse des Simulationsdatenmodells ausgerichtet ist. Dadurch konnte die Schnittstelle zielgerichtet entwickelt und realisiert werden.

Aufbauend auf der 3D-Konstruktion, dem SHD (Ship Hull Design)-Konstruktionsprozess und den Stücklistenreportern wurde eine Schnittstelle zur Simulation aufgebaut. Mit der erstellten Schnittstelle wurde die Möglichkeit geschaffen, Daten, die mit Hilfe eines Reporters (SHD Reporter) aus einem CATIA 3D Modell extrahiert wurden, im Produktmodell der Simulation zu hinterlegen. Dabei wird nicht nur eine "flache" Stückliste übergeben, sondern die Struktur der späteren Montage ("Hierarchy of Assembly") mit erzeugt (vgl. Abbildung 2) und an die Simulation übergeben. Des Weiteren werden alle Geometriedaten für das vereinfachte Acht-Punkte-Geometriemodell und die beschreibenden Attribute wie Länge, Breite, Höhe, Bauteiltyp etc. erzeugt. Abbildung 90 zeigt die "Hierarchy of Assembly", das vereinfachte Modell in

G:\Abschluss\Bericht\Gesamtbericht v03.11.doc

der Simulation und das reale Modell in CATIA. Beim Vergleich der realen in CAD erstellten Sektion mit dem vereinfachten Acht-Punkt-Geometriemodell wird deutlich, dass die Daten im Produktmodell in guter Qualität vorliegen und somit in ausreichender Genauigkeit für Simulationsanwendungen (Materialflusssimulation) zur Verfügung stehen. Die Bereitstellung von VRML-Dateien kann sowohl im VRML Format 1.0 und 2.0 ohne weitere Entwicklung erfolgen.

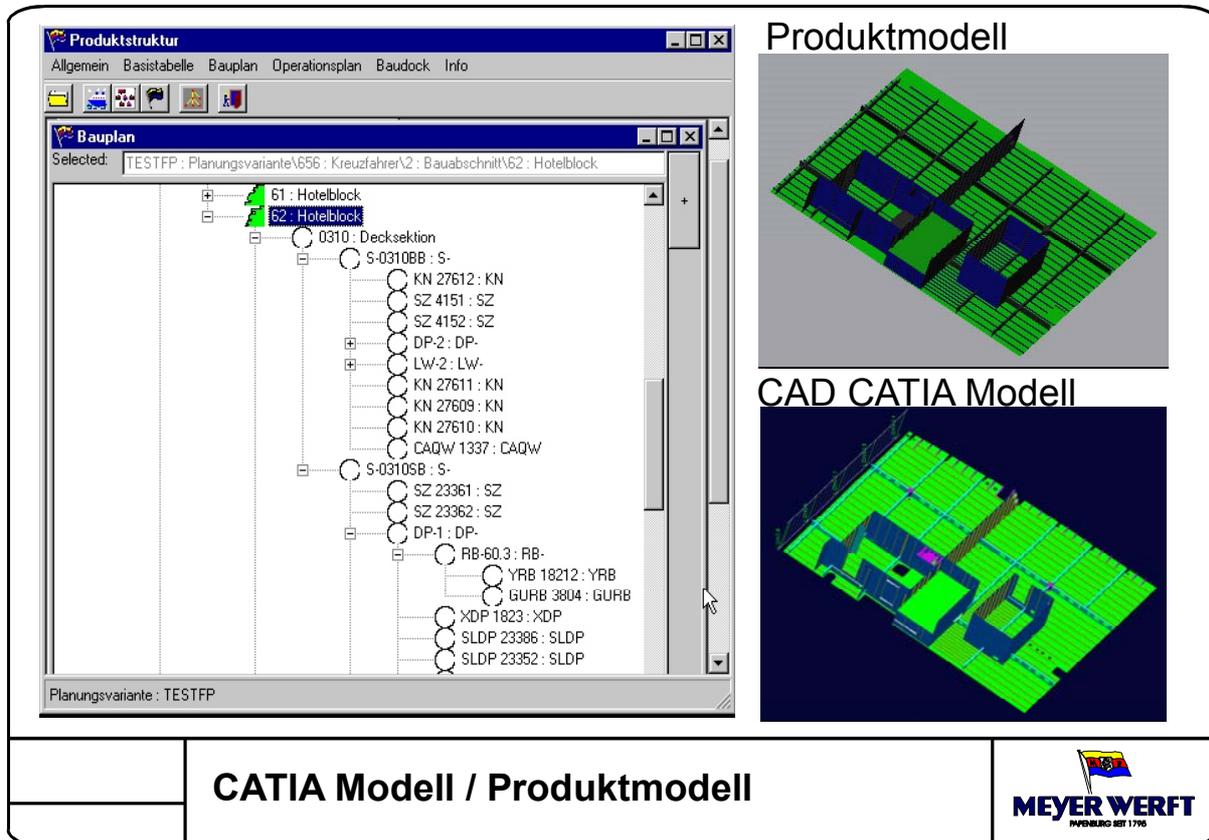


Abbildung 90 - CAD-Datentransfer: CATIA Modell / Produktmodell

Konstruktive Änderungen, die sich im Schiffskonstruktionsprozess ergeben, können mit der entwickelten Schnittstelle jederzeit für die Simulation bereitgestellt werden.

Durch die realisierte Schnittstelle wird das integrierte Datenmodell mit Daten aus einem Fremdsystem versorgt. Da die Schnittstelle vollständig auf die Anforderungen der Simulation ausgerichtet wurde, muss weder im SDM-Tool noch in den Bausteinen auf der Simulationsebene eine Erweiterung erfolgen. Im SDM-Tool können die CAD-Daten vollständig betrachtet und bearbeitet werden. Auf der Simulationsebene können die entsprechenden Bausteine (Datenbausteine) mit diesen CAD-Daten versorgt werden und so in allen Modellen universell eingesetzt werden.

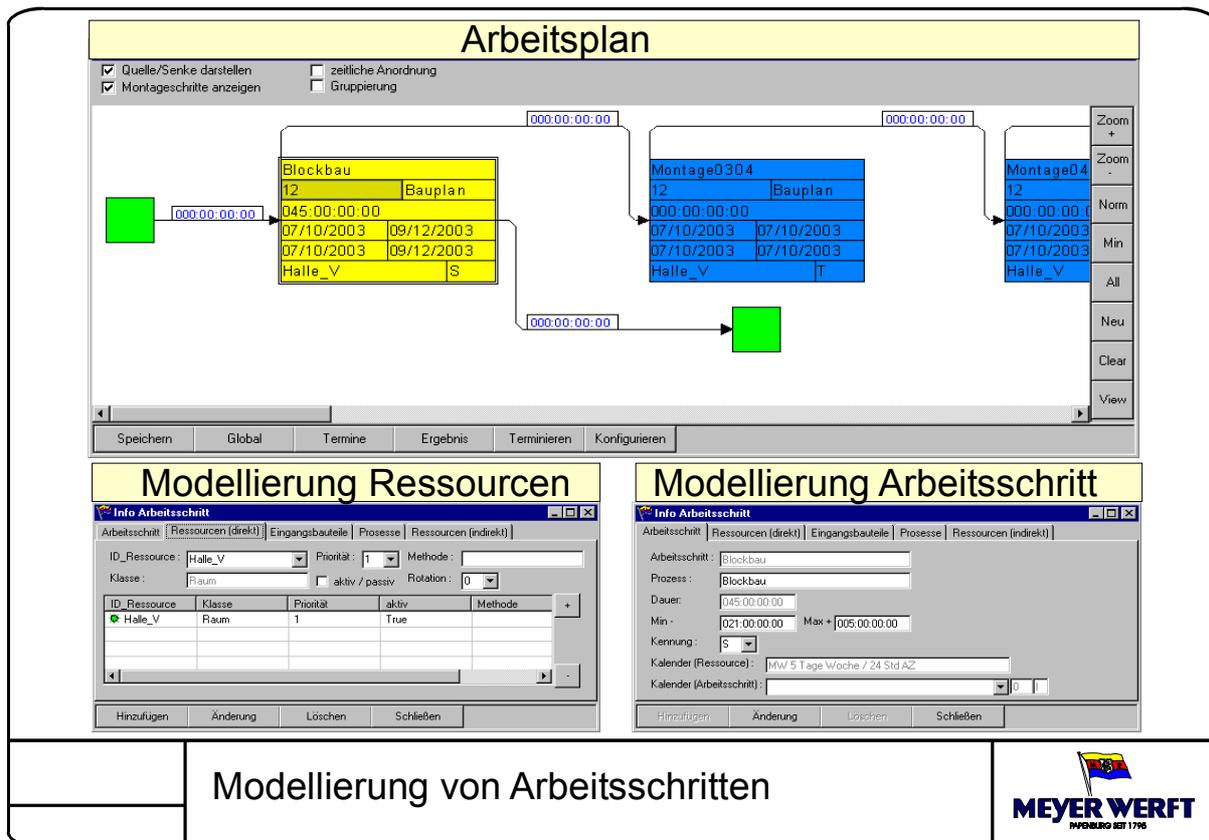
2.5.2 Datengenerierung

Für den Datenkomplex „Produkt“ wurden manuelle Eingabemöglichkeiten im SDM-Tool und eine Schnittstelle zum CAD-System CATIA realisiert. Für den Datenkomplex „Ressourcen“ wurden im SDM-Tool die entsprechenden Eingabemöglichkeiten geschaffen. Änderungen in diesem Datenkomplex sind selten erforderlich und können somit sehr gut manuell vom SDM-Tool gepflegt werden. Im Datenkomplex „Arbeitsplan“ werden die Informationen zum

Produkt mit den Informationen zu den Ressourcen in einem Arbeitsschritt verbunden. Abbildung 91 zeigt einen Arbeitsschritt in Netzplan.

2.5.2.1 Workflow

Besonders in diesem Datenkomplex entsteht ein sehr hoher Verwaltungsaufwand und ein sehr hohes Datenvolumen. Für den Anwender wurden Funktionen entwickelt, damit dieser Komplex flexibel verwaltet und bearbeitet werden kann. Im SDM-Tool wurde mit dem interaktiven Netzplan eine wichtige Komponente zur Datenpflege in das Toolset integriert (vgl. Querverweis). Die manuelle Erzeugung von individuellen Arbeitsplänen für jedes Bauteil würde für den Anwender jedoch einen unzumutbaren Aufwand bedeuten.



The screenshot displays the 'Arbeitsplan' (Network Plan) window at the top, which shows a flowchart of work steps. Below it are two 'Modellierung' (Modeling) windows: 'Modellierung Ressourcen' (Resource Modeling) and 'Modellierung Arbeitsschritt' (Work Step Modeling).

Arbeitsplan (Network Plan):

- Options: Quelle/Senke darstellen, Montageschritte anzeigen, zeitliche Anordnung, Gruppierung.
- Work Step 1 (Yellow):

Blockbau	12	Bauplan
045:00:00:00		
07/10/2003	09/12/2003	
07/10/2003	09/12/2003	Halle_V S
- Work Step 2 (Blue):

Montage0304	12	Bauplan
000:00:00:00		
07/10/2003	07/10/2003	
07/10/2003	07/10/2003	Halle_V T
- Work Step 3 (Blue):

Montage04	12	Bauplan
000:00:00:00		
07/10/2003	07/10/2003	
07/10/2003	07/10/2003	Halle_V T

Modellierung Ressourcen (Resource Modeling):

- Info Arbeitsschritt: Ressourcen (direkt) | Eingangsbauteile | Prozesse | Ressourcen (indirekt)
- ID_Ressource: Halle_V, Methode: 1
- Klasse: Raum, aktiv/passiv: aktiv, Rotation: 0

ID_Ressource	Klasse	Priorität	aktiv	Methode
Halle_V	Raum	1	True	

Modellierung Arbeitsschritt (Work Step Modeling):

- Info Arbeitsschritt: Ressourcen (direkt) | Eingangsbauteile | Prozesse | Ressourcen (indirekt)
- Arbeitsschritt: Blockbau
- Prozess: Blockbau
- Dauer: 045:00:00:00
- Min: 021:00:00:00, Max: 005:00:00:00
- Kennung: S
- Kalender (Ressource): Mw 5 Tage Woche / 24 Std AZ

Abbildung 91 - Datengenerierung: Interaktive Modellierung von Arbeitsschritten

In Simulationsprojekten werden die Produkte oftmals klassifiziert. Diesen Klassen werden dann verschiedene Arbeitspläne zugeordnet. Durch diese Vorgehensweise entsteht ein Datenmodell, in dem mit typisierten Arbeitsplänen gearbeitet wird. Für Simulationsanwendungen, die im Bereich der Serienfertigung eingesetzt werden, ist dies im allgemeinen eine gute und praktikable Vorgehensweise. In der ersten Projektphase wurde diese Vorgehensweise angewendet (vgl. Grobsimulation, 2.4.2.1). Sehr schnell zeigte sich jedoch, dass eine Klassifizierung des Produktes für eine Einzelfertigung nicht verwendet werden konnte. Damit eine realistische Abbildung im Modell erfolgen konnte, mussten sehr viele Klassen eingeführt werden, die oftmals nur minimale Abweichungen aufwiesen. Die Übersicht bei dieser Art von Datenmodell war oftmals sogar für den Simulationsexperten nicht gegeben. Dies führte im Projekt zur Entwicklung von individuellen Arbeitsplänen für alle Bauteile einer Variante.

2.5.2.2 Realisierung

Die Generierung von individuellen Arbeitsplänen basiert im wesentlichen auf dem oben beschriebenen Verfahren. In einem Standardarbeitsplan werden den unterschiedlichen Bauteiltypen Arbeitspläne hinterlegt. Aufgrund von Bauteilattributen (Klassen) wird bei der Datengenerierung für jedes Bauteil ein Arbeitsplan generiert und in der Datenbank gespeichert. Dieser Arbeitsplan entspricht einem in der Bibliothek abgelegtem Standardarbeitsplan. Der Anwender kann nun mit dem SDM-Tool den Arbeitsplan für jedes Bauteil bearbeiten und ihn so individuell gestalten. Diese erweiterte Vorgehensweise hat den Vorteil, dass nur wenige überschaubare Klassen und Standardarbeitspläne für die Datengenerierung erforderlich sind. Spezielle Anpassungen können somit ohne Anpassungen des Standards erfolgen. Als Nachteil muss die Größe des Datenmodells (Datenvolumen) erwähnt werden.

Die Erzeugung der Arbeitspläne erfolgt automatisiert durch den Einsatz eines Datengenerators, damit der Aufwand für die Dateneingabe für den Anwender gering und überschaubar gestaltet werden kann. Bei kompletter manueller Eingabe des Arbeitsplan für eine Variante entsteht ein Aufwand von mehreren Stunden bis zu mehreren Tagen für den Anwender. Durch den Einsatz eines Generators konnte dieser Aufwand auf wenige Minuten begrenzt werden.

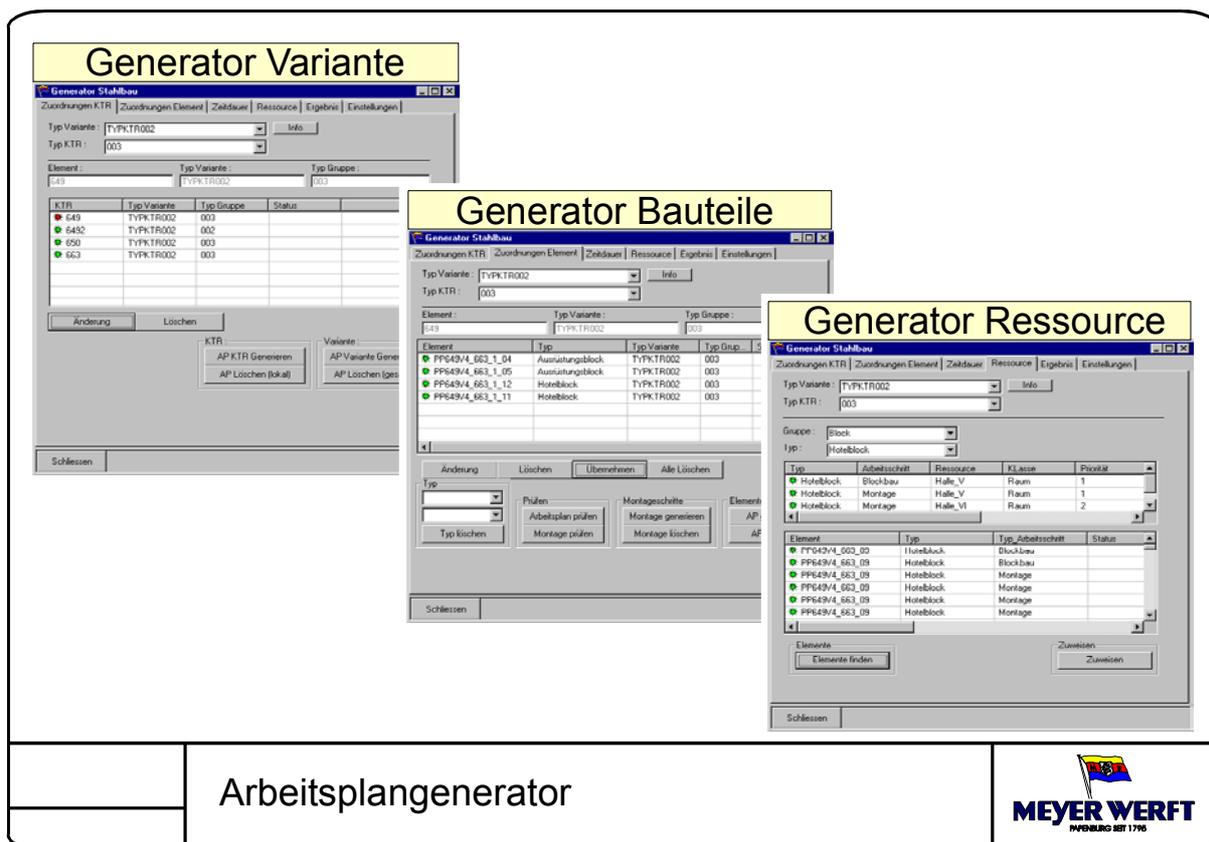


Abbildung 92 - Datengenerierung: Generator

In individuellen Arbeitsplänen wird für jedes Bauteil ein individueller Arbeitsplan im Datenmodell hinterlegt. Mit dem SDM-Tool kann dieser Arbeitsplan gepflegt und bearbeitet werden. Daher wurde auch die Funktion des Datengenerators in das SDM-Tool integriert. Für den Generator ergaben sich somit folgende Anforderungen:

- Generierung von individuellen Arbeitsplänen

Durch die Zuordnung von Schiffen zu den Standardarbeitsplänen soll für alle Bauteile eines Schiffes ein individueller Arbeitsplan erzeugt werden. Die Generierung kann für eine gesamte Variante (alle Schiffe), nur für ein Schiff einer Variante oder für ausgewählte Bauteile eines Schiffes erfolgen.

- Modifizierung des generierten Arbeitsplans
Arbeitspläne müssen oftmals modifiziert werden. Wenn für eine große Anzahl von Datensätzen Änderungen vorgenommen werden sollen wie z.B. Hinterlegung einer zusätzlichen alternativen Ressource im Arbeitsplan, muss dies mit dem Datengenerator möglich sein.
- Integration in das SDM-Tool
Der Datengenerator muss als ein Werkzeug im SDM-Tool integriert werden.
- Entwicklung und Abbildung von Standards in Bibliotheken
Standardarbeitspläne werden in Bibliotheken im Datenmodell abgelegt. Die Standardarbeitspläne können vom Anwender mit entsprechender Berechtigung bearbeitet bzw. neu angelegt werden. Der Zugriff auf diese Daten erfolgt mit dem SDM-Tool.

Die Entwicklung eines Generators hat wesentlich zum Erfolg der Simulationsanwendung beigetragen. Durch den Generator wird der Aufwand bei der Datenmodellierung drastisch gesenkt. Gleichzeitig wird für den Anwender ein sehr flexibles Datenmodell generiert. Individuelle Anpassungen können vom Anwender durchgeführt werden. In der Pilotphase wurde diese Funktion sehr gut angenommen. Abbildung 92 zeigt einige Funktionen des Datengenerators.

2.5.3 Optimierung

Im Simulationsmodell werden Regeln (Steuerungen) hinterlegt, die die im Simulationslauf lokal zu treffenden Entscheidungen und somit den Simulationslauf beeinflussen. In einem iterativen Prozess wird durch die Durchführung von Simulationsexperimenten, Analyse der Ergebnisse, Variation der Simulationsparameter und erneute Simulation ein Planungs- bzw. Simulationsoptimum erreicht.

Optimierungsverfahren sind mathematische Verfahren, mit deren Hilfe eine optimale Lösung für eine Planungsaufgabe ermittelt werden kann. Für bestimmte Anwendungsfälle stehen verschiedene Optimierungsverfahren zur Verfügung. Die Lösungsgüte (Qualität der Lösung) ist stark vom eingesetzten Verfahren abhängig. Optimierungsverfahren arbeiten mit einer globalen Zielfunktion. Die Zielfunktion muss für jede Aufgabenstellung neu formuliert bzw. angepasst werden. Da der Optimierungsalgorithmus die Güte einer Lösung aufgrund seiner Zielfunktion bewertet, muss die Zielfunktion besonders sorgfältig formuliert werden.

Im Projekt wurde der Einsatz von Optimierungsverfahren am Beispiel einer Flächengrobplanung untersucht. Das Beispiel wurde ausgewählt, um Vergleichsmöglichkeiten zur Simulation zu haben.

2.5.3.1 Implementierte Verfahren

Für die Flächengrobplanung stehen insgesamt drei Verfahren zur Verfügung: „Genetische Algorithmen“, „einstufiger Branch&Bound“ und „mehrstufiger Branch&Bound“.

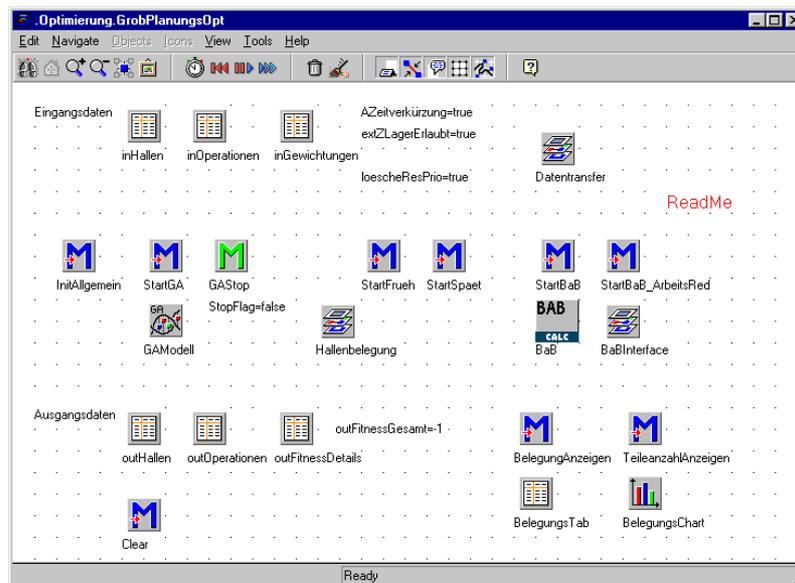


Abbildung 93 - Optimierung: Aufbau des Netzwerks

Alle relevanten Methoden und Netzwerke sind im Baustein „Optimierung“ zu finden. Abbildung 93 zeigt den Optimierungsbaustein. Der Baustein ist vollständig an das im Projekt entwickelte Datenmodell angebunden (Datenbaustein, Pausenbaustein etc.). Auswertungen können mit den beschriebenen Verfahren aus dem Simulationstoolset (z.B. Gantt / Bestand-Gantt, vgl. 2.2.7.1 f.) erledigt werden. Die Zielfunktion kann im Baustein parametrisiert werden.

Zusätzlich zu den drei Verfahren gibt es noch zwei einfache Heuristiken, die vornehmlich als Hilfsmittel der anderen Verfahren dienen. Mit der ersten Heuristik wird sukzessive für jede Operation der frühest mögliche Startzeitpunkt berechnet und nacheinander versucht, die Operation den möglichen Ressourcen ab diesem Zeitpunkt zuzuordnen. Ist dies für eine Operation nicht möglich, so wird die Operation der ersten Ressource unter Verletzung der Ressourcenkapazität zugewiesen. Mit der zweiten Heuristik wird analog der beschriebenen Heuristik für jede Operation versucht, sie zum spätest möglichen Zeitpunkt einer der erlaubten Ressourcen zuzuordnen.

Allen Verfahren ist gemein, dass sie die von ihnen gefundene Lösung in die Ergebnistabellen wegschreiben. Zusätzlich werden Qualität und Eigenschaften der gefundenen Lösung in einer Tabelle gespeichert. Zu den Einzelheiten sei auf den folgenden Abschnitt „Definition der Lösungsgüte (Fitness)“ verwiesen.

2.5.3.2 Definition der Lösungsgüte (Fitness)

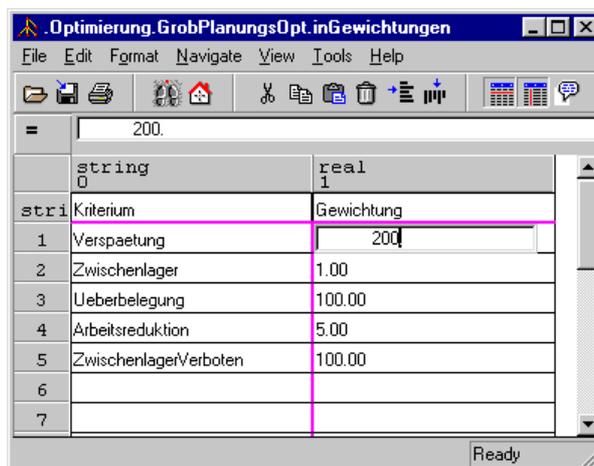
Das Ergebnis der Flächenplanung (die Lösung des Flächenplanungsproblems also) besteht in der Zuordnung von Operationen auf die einzelnen Ressourcen sowie in der Zuordnung von Startzeitpunkten für jede Operation. Die Qualität einer solchen Lösung (Fitness) wird anhand mehrerer Kriterien beschrieben:

a) Verspätung

Für jede Operation gibt es einen vorgegebenen spätesten Fertigstellungstermin. Dieses Kriterium summiert für alle Operationen, bei denen das Ende den spätesten Termin überschreitet, die Differenz dieser Termine in Tagen auf.

- b) **Überbelegung**
Jede der Hallen hat eine vorgegebene Kapazität. Die Summe des Platzbedarfes aller Teile in einer Halle sollte zu keinem Zeitpunkt größer sein als diese Hallenkapazität. Sollte es trotzdem notwendig sein, zu viele Teile einer Halle zuzuordnen, so gibt dieses Kriterium ein Maß für die Verletzung der Hallenkapazität an. Über alle Tage wird die zuviel in Anspruch genommene Kapazität in Tagen berechnet.
- c) **Arbeitsreduktion**
Jede Operation besitzt eine Bearbeitungszeit/Dauer. Zusätzlich kann für jede Operation angegeben werden, ob diese Dauer bei Bedarf reduziert werden kann. Das Kriterium der Arbeitsreduktion summiert die Verkürzungen der Bearbeitungszeiten aller Operationen zusammen und misst diese in Tagen.
- d) **Zwischenlager**
Dieses Kriterium umfasst die Anzahl von Tagen, die von Operationen als Zwischenlagerung benutzt werden.
- e) **ZwischenlagerVerboten**
Dieses Kriterium umfasst die Anzahl von Tagen, die Teile extern zwischengelagert werden.

Jedes dieser fünf Kriterien wird gewichtet. Die festgelegte Wichtigkeit jedes Kriteriums wird durch einen Gewichtungsfaktor beschrieben. Dabei steht eine hohe Gewichtung für ein wichtiges Kriterium. Abbildung 94 zeigt eine Einstellung, bei der den Kriterien „Zwischenlager“ und „Arbeitsreduktion“ geringes Gewicht beigemessen wird.



string 0	real 1
Kriterium	Gewichtung
1 Verspaetung	200
2 Zwischenlager	1.00
3 Ueberbelegung	100.00
4 Arbeitsreduktion	5.00
5 ZwischenlagerVerboten	100.00
6	
7	

Abbildung 94 – Optimierung: Fitness

Um die Qualität der Lösung auf einen einzigen Zahlenwert zu reduzieren, wird die gewichtete Summe über alle Kriterien gebildet. Das Ergebnis wird Fitness genannt. Dieser Begriff ist dem Vokabular der Genetischen Algorithmen entnommen. Im vorliegenden Fall steht ein kleiner Fitness-Wert für eine gute Lösung, ein großer Fitness-Wert entsprechend für eine vergleichsweise schlechte Lösung. Fitness-Werte erlauben es, die Qualität unterschiedlicher Lösungen einfach zu vergleichen.

2.5.3.3 Optimierungsverfahren

Dieser Abschnitt behandelt die generelle Vorgehensweise bei der Optimierung. Es wird erläutert, in welcher Reihenfolge die Verfahren eingesetzt werden sollten. Die spezifischen Eigenschaften der implementierten Verfahren sind in den folgenden Kapiteln beschrieben.

Bei jedem neuen Datensatz sollte zuerst versucht werden, den einstufigen Branch&Bound zu starten. Dieses Verfahren liefert eine Lösung guter Güte, falls es eine Lösung findet. Sollte dieses Verfahren keine Lösung finden, so ist das mehrstufige Branch&Bound zu versuchen. Das mehrstufige Branch&Bound wird in der Regel häufiger eine Lösung finden als der einstufige Branch&Bound, allerdings ist die Lösungsgüte im allgemeinen schlechter. Sollte auch das mehrstufige Branch&Bound keine Lösung liefern, so sind die Genetischen Algorithmen zu verwenden. Diese finden in jedem Falle eine Lösung, deren Qualität im Vergleich zu den Branch&Bound-Verfahren wiederum schlechter ist.

Durch die Existenz der drei implementierten Verfahren ist eine differenzierte Behandlung von „unterschiedlich schwierigen“ Probleminstanzen möglich.

Einstufiger Branch&Bound

Die implementierten Branch&Bound-Verfahren gehören zu den Constraint-basierten Ansätzen. Dabei werden die geforderten Eigenschaften einer Lösung mittels Constraints (Bedingungen) definiert. Beim vorliegenden Problem handelt es um zeitliche Bezüge zwischen Operationen (Vorgänger- und Nachfolgerbeziehungen) oder genauer zwischen deren Start- und Endzeitpunkten. Zusätzlich wird die maximale Kapazität der Hallen berücksichtigt.

Die aufgestellten Constraints stellen strikte Bedingungen dar, die vom Verfahren unter keinen Umständen verletzt werden. Sind die Constraints zum Beispiel widersprüchlich, so beendet das Verfahren die Suche mit dem Ergebnis, dass keine Lösung gefunden werden konnte.

Das implementierte Suchverfahren ist aus Gründen der Effizienz keine vollständige Suche. Das heißt, dass sich die Suche unter der Menge der potentiellen Lösungen auf eine Teilmenge von aussichtsreichen Kandidaten beschränkt.

Das Branch&Bound-Verfahren versucht nacheinander, allen Operationen einen Startzeitpunkt und eine Ressource zuzuordnen. Dabei wird zuerst geprüft, ob die entsprechende Ressource genügend freie Kapazität für den gewünschten Zeitraum beinhaltet. Anschließend werden die zeitlichen Beziehungen zwischen den Operationen auf Erfüllbarkeit untersucht. Dabei wird stets auch vorausschauend nachgesehen, ob die Zuweisung aller noch nicht zugewiesenen Operationen möglich bleibt.

Stellt das Verfahren fest, dass keine weitere Zuordnung möglich ist, so müssen bereits vollzogene Zuordnungen von Operationen auf Ressourcen zurückgenommen werden. Diesen Vorgang nennt man auch *backtrack*. Um nicht alle möglichen Lösungen untersuchen zu müssen, merkt sich das Verfahren während der Suche aussichtsreiche, alternative Startzeiten für bereits zugewiesene Operationen. Auf diese Alternativen wird im Falle einer „Sackgasse“, wenn also keine weitere Operation mehr zugewiesen werden kann, zurückgegriffen.

Die Reihenfolge, in der das Branch&Bound-Verfahren die Ressourcen und Zeitpunkte durchsucht, ist individuell für jede Operation steuerbar. Dies geschieht über Prioritäten, die den alternativen Ressourcen einer Operation zugewiesen werden. Haben alle Ressourcen dieselbe Priorität, so werden zu einem gegebenen Zeitpunkt nacheinander alle erlaubten Ressourcen versucht zu belegen, bevor in der Zeit fortgeschritten wird. Es wird jeweils am Startzeitpunkt festgehalten und die Ressourcen variiert.

Unterscheiden sich die Prioritäten der Ressourcen einer Operation, so wird stets bevorzugt versucht, die erste Ressource zu belegen. Auf dieser Ressource werden die in Fragen kommenden Startzeitpunkte durchgespielt und erst dann auf die nächste Ressource zurückgegriffen. In diesem Fall wird also an einer Ressource festgehalten, die Startzeitpunkte werden variiert.

Eigenschaften des Branch&Bound-Verfahrens

Die Stärke der constraintbasierten Suchverfahren liegt in der Behandlung von Problemstellungen, in denen die vorgegebenen Randbedingungen strikt einzuhalten sind. Aus diesem Grunde eignet sich der einstufige Branch&Bound vor allem für die Suche nach „annähernd perfekten“ Lösungen, die keine der Randbedingungen verletzen. Kann davon ausgegangen werden, dass eine solche Lösung existiert, dann ist dieses Verfahren besonders geeignet.

Zusätzlich eignet sich das Branch&Bound-Verfahren dazu, um zu untersuchen, ob es eine „perfekte“ Lösung gibt. Da das implementierte Verfahren keine vollständige Suche ist, sondern nur eine Teilmenge der potentiellen Lösungen untersucht, ist ein Fehlschlagen des Branch&Bound-Algorithmus keine Gewähr, aber ein gutes Indiz dafür, dass es eine Lösung ohne Verletzung mindestens einer der Randbedingungen nicht gibt.

Die Schwäche des Verfahrens (z.B. im Vergleich mit der implementierten Optimierung mittels Genetischen Algorithmen) liegt darin, dass es vorkommen kann, dass die Suche erfolglos endet. In diesem Falle ist aufgrund des constraintbasierten Ansatzes die Ausgabe einer Lösung prinzipiell unmöglich.

Aus den beschriebenen Gründen wird vorgeschlagen, die Suche nach einer Lösung für einen neuen Datensatz mit dem einstufigen Branch&Bound beginnen zu lassen.

Die Branch&Bound-Suche ist eine systematische Suche und endet in jedem Fall nach endlicher Zeit. Im allgemeinen ist der Zeitbedarf für diese Suche (für z.B. einige Hundert Operationen) allerdings viel zu groß, so dass ein Kriterium zum vorzeitigen Abbrechen definiert werden muss. Im implementierten Verfahren ist dies die Anzahl der maximal vollzogenen Backtracks.

Mehrstufiger Branch&Bound

Die grundsätzliche Idee hinter dem mehrstufigem Branch&Bound liegt darin, die Constraints der Branch&Bound-Suche zielgerichtet zu lockern und anschließend die Verletzung der Randbedingungen schrittweise zurückzunehmen. Aus diesem Grunde ist die Anwendung des mehrstufigen Branch&Bound nur dann sinnvoll, wenn der originale, im vorangehenden Kapitel beschriebene einstufige Branch&Bound keine Lösung findet.

Der mehrstufige Branch&Bound setzt sich aus mehreren Optimierungsschritten zusammen:

1. Im ersten Schritt wird das originale Branch&Bound-Verfahren dahingehend aufgeweicht, dass im Falle von hartnäckigen Engpässen eine Überbelegung von Ressourcen erlaubt wird. Das Ergebnis des ersten Laufes ist eine Belegung der Ressourcen, die zu einigen Zeitpunkten Überbelegungen aufweist. Im zweiten Schritt wird für jede Ressource individuell der erste und letzte Tag bestimmt, an dem eine Überbelegung auftritt. Nun wird allen Operationen, die innerhalb dieses Zeitintervalls liegen, ihre jeweils maximal erlaubte Arbeitsreduktion zugewiesen. Operationen, denen keine Arbeitsreduktion erlaubt ist, bleiben unverändert.
2. Im dritten Schritt wird erneut das Branch&Bound-Verfahren gestartet. Dabei werden die im zweiten Schritt festgelegten Arbeitsreduktionen benutzt. Überbelegungen der Ressourcen werden in diesem Lauf nicht erzeugt; die Kapazitäten der Ressourcen werden strikt eingehalten. Findet das Branch&Bound-Verfahren in diesem Schritt keine Lösung, so ist der mehrstufige Branch&Bound gescheitert. In diesem Fall ist das Maß der erlaubten Arbeitsreduktionen zu gering, um die im ersten Schritt benötigten Überbelegungen auszugleichen.

3. Zuletzt wird versucht, die im dritten Schritt benutzten Arbeitsreduktionen schrittweise zurückzunehmen. Zur Entscheidung, ob eine Arbeitsreduktion für eine Operation zurückgenommen werden darf oder nicht, werden die Constraints des Branch&Bound auf Einhaltung untersucht.

Die Eigenschaften des mehrstufigen Branch&Bound-Verfahrens hängen naturgemäß eng mit denen des einfachen Branch&Bound-Verfahrens zusammen. Es ist anzumerken, dass jede der beiden Branch&Bound-Läufe (Schritte 1 und 2) scheitern können. Sollte dies der Fall sein, so ist auch das mehrstufige Branch&Bound-Verfahren gescheitert.

Genetische Algorithmen

Genetische Algorithmen generieren in einem iterativen Prozess Lösungsvorschläge für ein Optimierungsproblem. Die Anwendung bestimmt die Güte der Lösungsvorschläge (Fitness-Wert). Diese Güte entscheidet, ob bestimmte Lösungsvorschläge zur Generierung neuer Lösungsvorschläge herangezogen werden oder nicht.

Genetische Algorithmen versuchen die Zusammenhänge der Evolution in der Natur auf die Suche nach der Lösung eines Optimierungsproblems zu übertragen. Sie benutzen das bewährte System der Evolution. Aus einer Startmenge von Lösungsvorschlägen, werden die „erfolgreichsten“ ausgewählt, und aus diesen neue Lösungsvorschläge erzeugt. Danach erscheinen in einem zyklischen Prozess immer neue, bessere Generationen von Lösungsvorschlägen. Ob ein Lösungsvorschlag einen Selektionsvorteil besitzt und zur Generierung neuer Lösungsvorschläge herangezogen werden soll, hängt vom sogenannten Fitness-Wert ab, der von der jeweiligen Anwendung berechnet wird.

Genetische Algorithmen betrachten also stets eine Menge von Lösungen (*Individuen*) parallel. Die Menge der momentanen Lösungen wird entsprechend Generation genannt. Aus jeder Generation werden neue Lösungen gebildet, aus denen dann die Elemente einer neuen Generation gebildet werden.

Wie bilden genetische Algorithmen neue Lösungen?

Nachdem der genetische Algorithmus die Startgeneration der Individuen generiert hat, führt er den Generationszyklus durch. Der Generationszyklus besteht aus...

- ...der stochastischen oder deterministischen Auswahl der Eltern (Reproduktion der Lösungen),
- ...der Generierung der Nachkommen (Kinder) generiert durch die zufällige Anwendung der genetischen Operatoren (Crossover, Mutation, Inversion) und
- ...der Berechnung der Fitness-Werte der Nachkommen (Zielfunktion)

Nach der Initialisierung werden die Individuen der Startpopulation bewertet, indem für jede Lösung ein Fitness-Wert ermittelt wird. Danach bestimmt eine Selektionsregel (Auswahl der Eltern), welche Eltern aus der Startpopulation ausgewählt werden. Aus den ausgewählten Individuen werden nun Elternpaare gebildet. Beide Eltern werden kopiert, so dass Sohn und Tochter Klone des Vater bzw. der Mutter sind. Danach werden auf Sohn und Tochter die genetischen Operatoren angewandt. Die Bewertungsmethode startet dann die Berechnung des Fitness-Wertes.

Aus jeder Familie wird, entsprechend der Einstellung des Parameters Auswahl der Nachkommen, das Individuum für die nächste Generation ausgewählt. Sind alle Individuen für die nächste Generation ermittelt, werden aus diesen Individuen neue Elternpaare ermittelt, aus denen die Kinder für die nächste Generation erzeugt werden, etc.

Parameter und Eigenschaften des Verfahrens

Das Anwendungsgebiet von Genetischen Algorithmen (GA) liegt hauptsächlich bei Optimierungsaufgaben, die folgende Eigenschaften haben:

- Große Lösungsräume.
- Unbekannte Eigenschaften des Lösungsraumes.
- Lösungsräume mit Diskontinuitäten, die eine mathematisch-numerische Optimierung des Problems nicht zulassen.
- Viele weiche Restriktionen der Lösungen. Unter weichen Restriktionen versteht man Einschränkungen, die nicht zwingend eingehalten werden müssen, die jedoch zu Verschlechterungen führen.

Ein prinzipieller Vorteil der Genetischen Algorithmen liegt darin, dass man die Suche zu einem beliebigen Zeitpunkt abbrechen und dennoch stets auf eine gültige Lösung zugreifen kann.

Im vorliegenden Fall werden die Genetischen Algorithmen benutzt, um Startzeiten, zu belegende Ressource und ggf. Arbeitsreduktion der Operationen festzulegen. Anhand dieser Daten werden die Operationen dann so spät wie möglich den Ressourcen zugewiesen.

Ein wichtiger Parameter bei der Konfiguration von Genetischen Algorithmen ist die Größe der Generation. Dies ist die Anzahl der parallel betrachteten Lösungen. Wird dieser Wert zu groß gewählt, so dauert das Bearbeiten einer Generation sehr lange. Wird der Parameter zu klein gewählt, so können zwar viele Generationen in kurzer Zeit bearbeitet werden, aber das Spektrum an verschiedenen Lösungen ist dann in der Regel zu klein, als dass eine sehr gute Lösung gefunden werden kann.

Die Genetischen Algorithmen sind ein Verfahren, welches prinzipiell beliebig lang laufen kann. Aus diesem Grunde muss ein relativ willkürlicher Termin für das Beenden der Optimierung festgelegt werden. Üblicherweise geschieht dies über die Angabe einer maximalen Anzahl der Generationen, die gerechnet werden sollen.

Nach Möglichkeit sollte man die Genetischen Algorithmen so lange laufen lassen wie möglich. Bei Bedarf kann man jedoch die Suche abbrechen, wenn sich die Qualität der besten Lösung schon einige Generationen nicht verbessert hat und zusätzlich das Spektrum der vertretenen Fitness-Werte (schlechtester Fitness-Wert zu bester Fitness-Wert) schmal ist. In dem Fall wird die jeweils aktuelle Generation zu Ende bearbeitet und erst dann die Optimierung beendet.

Während der Optimierung informiert ein Diagramm über den Fortschritt (Abbildung 95). Dem Diagramm kann für jede Generation der beste, der durchschnittliche und der schlechteste Fitness-Wert unter den Lösungen einer Generation entnommen werden.

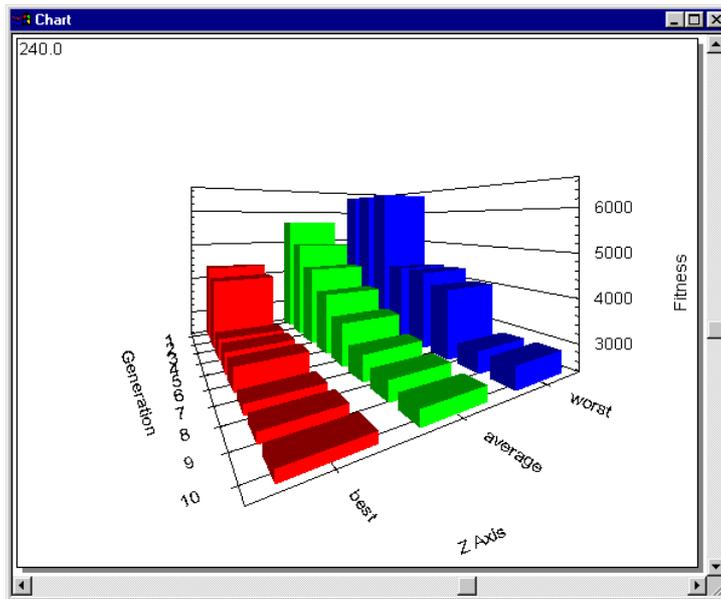


Abbildung 95 - Optimierung: Entwicklung der Fitnesswerte

Ergebnisse und Vergleich

Für den Test der Verfahren wurde eine Vergleichsstudie zwischen der Flächen-Grobsimulation und der Flächenoptimierung durchgeführt. Basis für den Vergleich war das Ressourcenmodell aus der Flächenbelegung (vgl. 2.4.1.2). Im Modell wurden die Hallen der Sektionsvorfertigung Halle II – Halle IV, die Halle der Blockfertigung Halle V –VI und die Baudocks I-II berücksichtigt. Beide Verfahren (Optimierung / Simulation) wurden mit den gleichen Daten versorgt, um einen direkten Vergleich zu ermöglichen. In Abbildung 96 und Abbildung 97 wurden die Ergebnisse zusammengefasst.

Beide Verfahren liefern gute Ergebnisse. Die modellierten Strategien, Regeln im Simulationsmodell und Zielfunktion in der Optimierung haben zu einer guten Planungslösung geführt. Alternative Ressourcen wurden von beiden Verfahren bei der Lösungsfindung herangezogen. Die Abbildungen zeigen einen Vergleich der Simulations- und Optimierungsergebnisse mit den Ergebnissen einer infiniten Kapazitätsplanung (Just in Time). Die Möglichkeit der Arbeitsreduktionen wurden vom Optimierungsmodell verwendet, Zwischenlagerungen wurden vom Simulationsmodell verwendet. In beiden Lösungen ergeben sich nur minimale Unterschiede.

In Abbildung 97 werden die Ergebnisse des zweistufigen Branch&Bound dargestellt.

Vergleich Simulation - Optimierung:

Im Projekt zeigte sich, dass die Definition der Zielfunktion und die Modellierung mit sehr hohem Aufwand verbunden sind und spezielles “Optimierung- Know How” erfordern. Des Weiteren ist im Optimierungsverfahren wenig Transparenz vorhanden, so dass nur wenig Akzeptanz für die gefundene Lösung besteht. Die Entwicklung und der Einsatz von individuellen Optimierungsverfahren mit sehr hohem Aufwand und sehr viel “Optimierungs- Know How” verbunden ist. Tabelle 13 zeigt einen Vergleich zwischen Simulations- und Optimierungsverfahren.

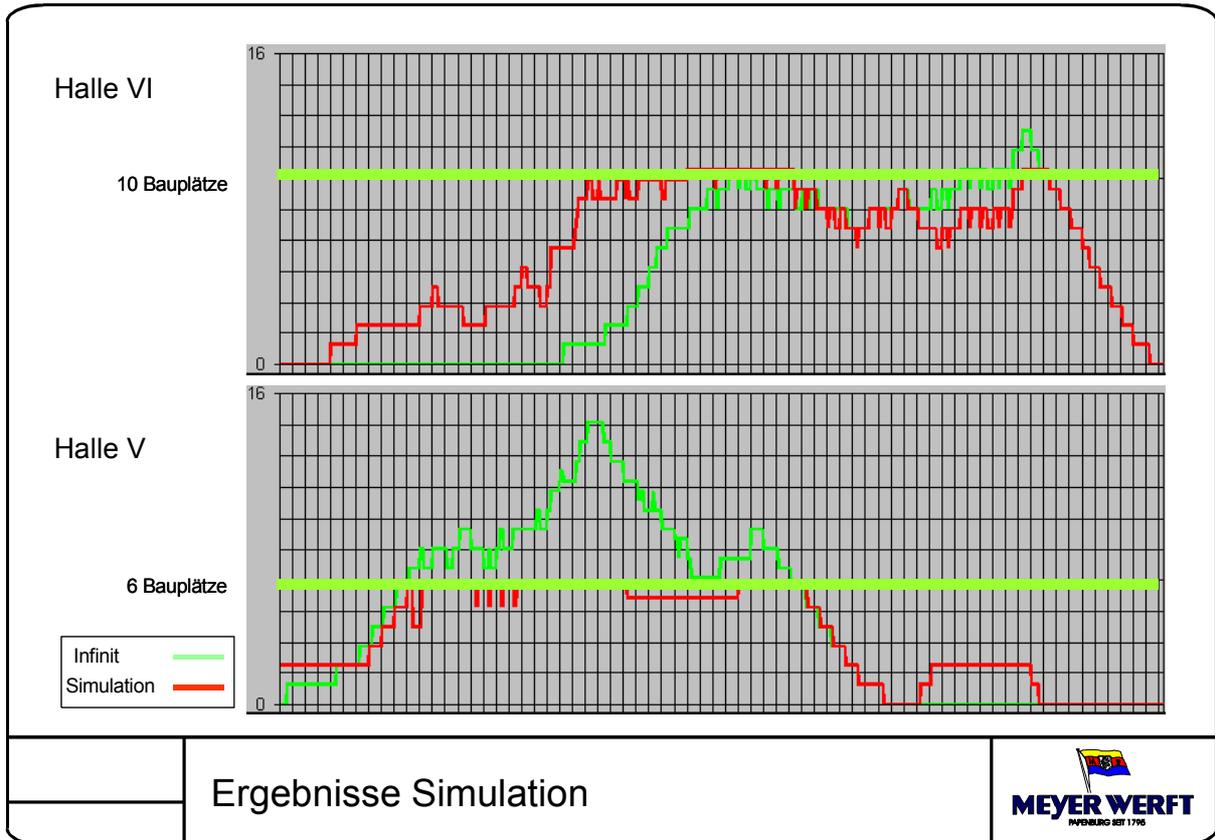


Abbildung 96 - Optimierung: Ergebnisse Simulation

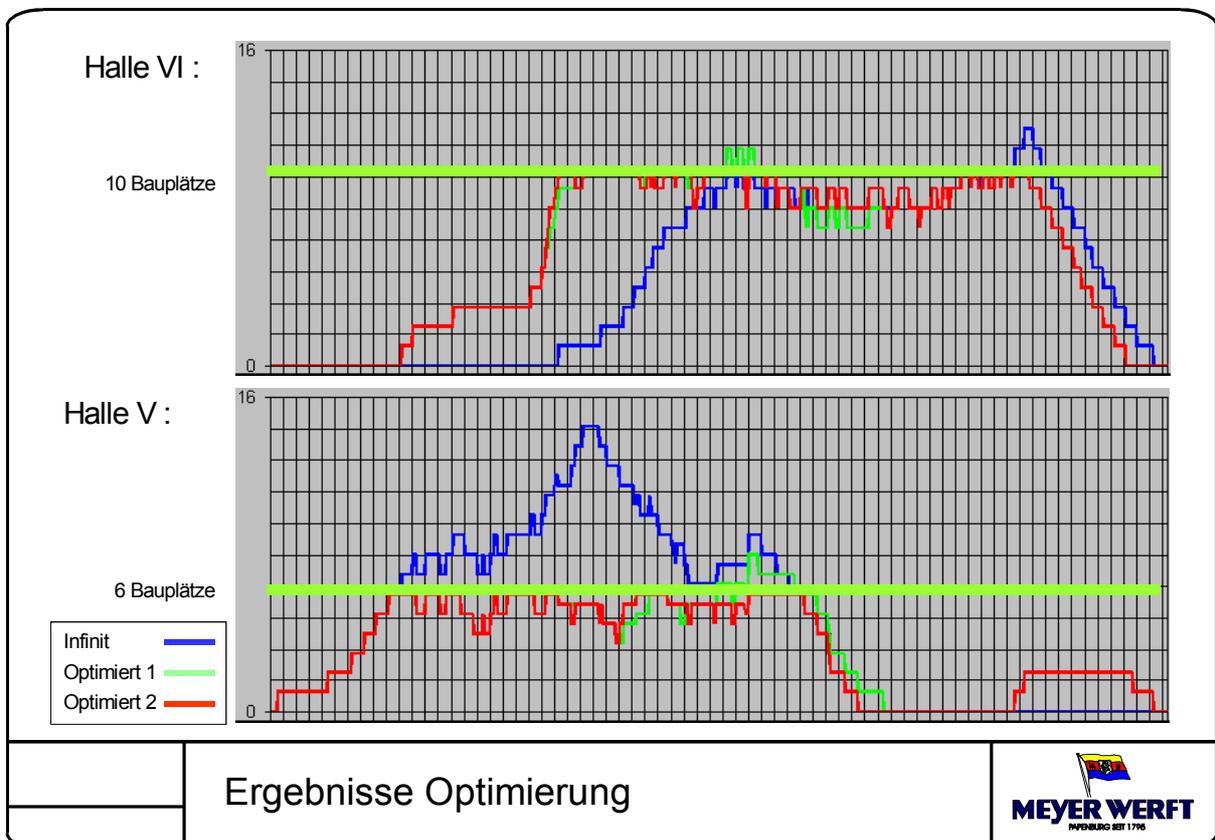


Abbildung 97 - Optimierung: Ergebnisse Optimierung

G:\Abschluss\Bericht\Gesamtbericht v03.11.doc

	Optimierung	Simulation
Transparenz	Niedrig	Hoch
Zielverfolgung	Global	Lokal
Steuerung	Zielfunktion	Regeln
Modellierung	sehr hoher Aufwand	mittlerer Aufwand
Experimente	nicht erforderlich	erforderlich

Tabelle 13 - Optimierung: Kriterien und Vergleich von Optimierung und Simulation

Die Modellierung einer guten Simulationslösung ist auch mit geringem „Simulations- Know How“ möglich. Der Einsatz der unterschiedlichen Verfahren ist jeweils individuell abzuwägen.

2.5.4 Bausteinverwaltung und Dokumentation

Die fortschreitende Entwicklung des Projektes bedingt eine Vielzahl von unterschiedlichen Methoden, Bausteinen und Modellen. Zur Dokumentation der erstellten Methoden, Bausteinen und Simulationsmodelle wurde ein Dokumentationskonzept entwickelt. Zur vollständigen Dokumentation des Simulationstoolsets müssen auf der untersten Ebene die Methoden (Code), auf der mittleren Eben die Bausteine (Objekte) und auf der höchsten Ebene die Modelle dokumentiert werden. Für alle drei Ebenen wurde ein Konzept mit folgenden Bestandteilen entwickelt.

Methodendokumentation (unterste Ebene):

- Method Manager
- Dokumentationsrichtlinie

Bausteindokumentation (mittlere Ebene):

- Baustein und Objekt Datenbank
- Dokumentationsrichtlinie

Modelldokumentation (höchste Ebene):

- Anwendungsdokumentation /Schulungsunterlagen
- Dokumentationsrichtlinie

Da das Simulationstoolset von Anwendern in den Fachabteilungen eingesetzt werden soll, wurde in anbetracht des steigenden Umfangs des Toolsets eine Verwaltung der Bausteine und Objekte vorgesehen. Die Administration der erstellten Klassen soll eine Versionskontrolle und das Auffinden existierender Komponenten erleichtern. Daher ist innerhalb des Projektes mit der Entwicklung der hier beschriebenen Anwendung begonnen worden, in der Versionsunterschiede der Komponenten gepflegt werden können. Vor allem die Kontrolle unterschiedlicher Versionen von Bausteinen erwies sich als erforderlich, da im Projekt gleichzeitig mehrere Entwickler eingebunden waren und eine Versionsverwaltung einzelner Objekte in eM-Plant nicht vorgesehen ist.

Neben einer Übersicht über die verschiedenen Entwicklungsstände soll die Anwendung ein schnelles Auffinden geeigneter Objekte ermöglichen. Hierzu werden einem Simulationsbaustein vom Entwickler Stichwörter zugeordnet.

Die Anwendung zielt auf eine Integration in die Simulationsumgebung. Über das in Simulationsbausteinen instanziierte „Dokumentationsobjekt“ (vgl. 2.2.2.1) können Dokumentationen mit Hilfe der eM-Plant eigenen ODBC-Schnittstelle in den Datenbestand aufgenommen werden. Die Pflege der Daten erfolgt mit dem im Projekt entwickelten Programm Bausteinverwaltung.

2.5.4.1 Anforderungen „Dokumentation“

Ausgehend von einer allgemeinen Beschreibung eines Bausteins ermöglicht die Modellierung, Abhängigkeiten zu Simulationsmodellen und anderen Bausteinen aufzuzeigen. Diese Funktionalität unterstützt den Entwickler der Bausteinbibliothek, wenn er bestehende Methoden eines Bausteins effizienter implementiert hat oder die Schnittstellen öffentlicher Methoden verändert hat.

Öffentliche Methoden bilden die Schnittstelle des erstellten Objektes zur umgebenden Simulationsanwendung. Andere Entwickler können diese Methoden aufrufen, um die Funktionalität des Bausteins zu verwenden. Daher sind allen Entwicklern die Schnittstellen bekannt zu machen. Die Dokumentation der Aufrufe wird in der Datenbank in der Datentabelle „Methode“ abgelegt.

Zumal ein Baustein iterativen Entwicklungsprozessen unterliegt, sollten Aussagen über Versionsunterschiede möglich sein. Versionen und Beziehungen zwischen den Versionen und neu erstellten Methoden können dokumentiert werden.

Die Zahl der in der Datenbank hinterlegten Bausteine ist im Verlauf des Projektes stetig angestiegen. Daher ist ein schnelles Auffinden geeigneter Bausteine über Suchkriterien zu unterstützen. Hierzu werden Stichwörter zu Bausteinen und Modellen verwaltet.

Neben der Dokumentation der vorgenommenen Änderungen ist ein einfaches Erfassen geplanter Änderungen erforderlich. Zur Übersicht noch durchzuführender Arbeiten an einem Baustein können „ToDo“-Arbeitspunkte und Informationen über geplante und bereits abgeschlossene Änderungen dokumentiert werden.

2.5.4.2 Bausteinverwaltung

Die Anwendung „Bausteinverwaltung“ besteht aus unabhängigen Dialogen. Die Dialoge werden innerhalb eines übergeordneten Programmablaufes verwaltet. Im folgenden werden die Dialoge zur Pflege der Datenbasis an Hand von Bildschirmdarstellungen eingehender beschrieben.

Erfassen von Bausteinen

Der Dialog „Bausteine Erfassen“ ist in Abbildung 98 dargestellt. Es werden alle Datenfelder der Tabelle „Baustein“ angezeigt. Des Weiteren sind Informationen über übergeordnete Modelle und abhängige Bausteine zu entnehmen.

Der Anwender kann die Inhalte der Eingabefelder ändern. Nach Auswahl der Funktion „Anlegen“ werden die Werte in die zugeordnete Tabelle übernommen. In der Datenbank ist ein Baustein eindeutig über seine Bezeichnung identifiziert. Sind bereits Daten zu einem Baustein erfasst, werden die Werte aktualisiert.

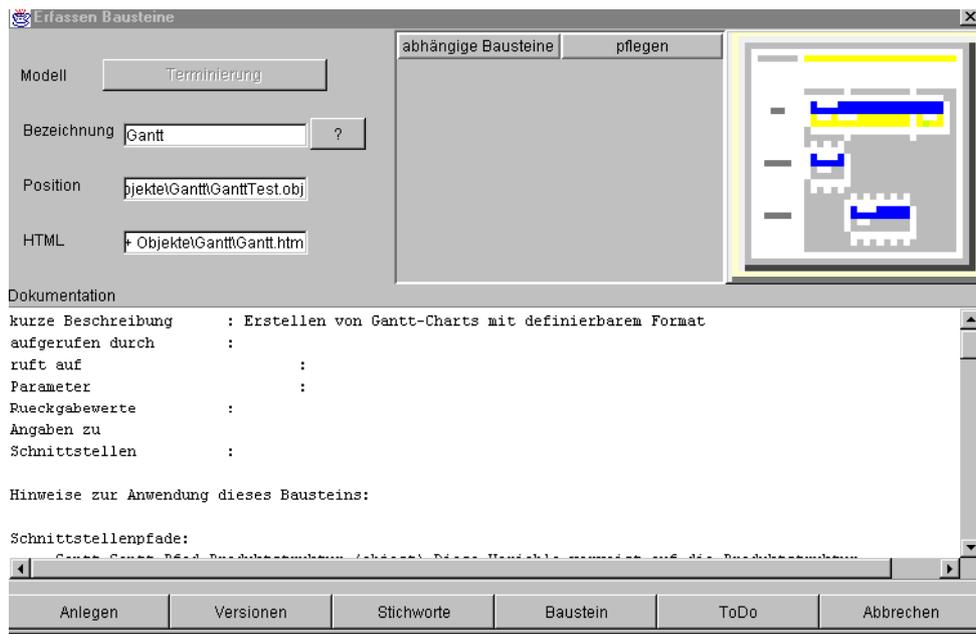


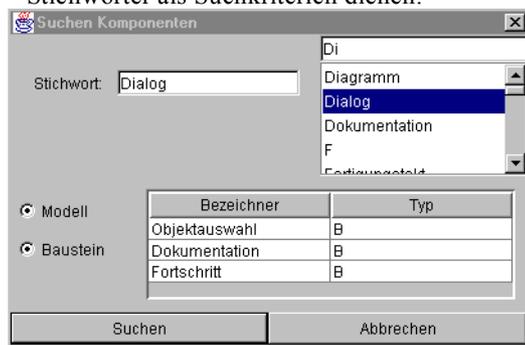
Abbildung 98 - Bausteinverwaltung: Dialog „Erfassen Bausteine“

Abhängigkeiten zwischen zwei Bausteinen können ebenso erfasst und verwaltet werden. Die Abhängigkeit zwischen den Bausteinen entsteht, wenn ein Baustein in einen anderen transferiert wird, um die Funktionalität zu nutzen.

Suchfunktionen

Suchen von Komponenten

Auffinden von Bausteinen, wobei Stichwörter als Suchkriterien dienen.



Erfassen der Suchkriterien

Erfassen von Stichwörtern, mit deren Hilfe ein Baustein in der Datenbank gesucht werden kann.

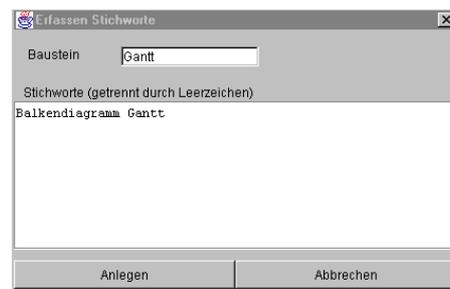


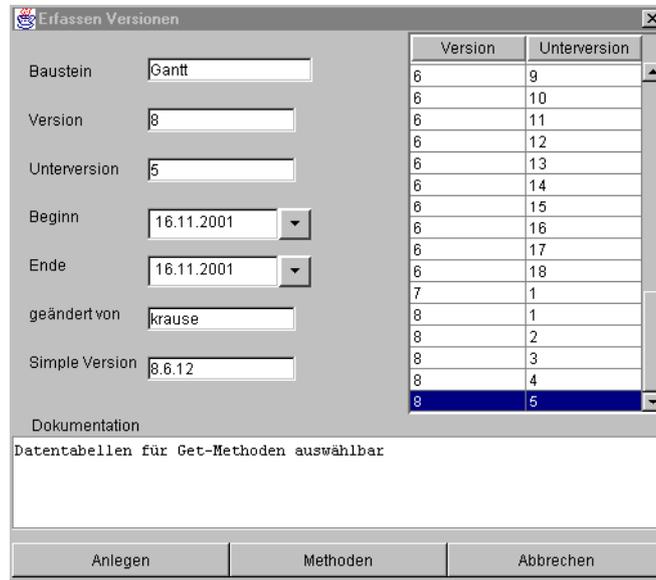
Abbildung 99 - Bausteinverwaltung: Dialoge „Suchen Komponenten“ (links) und „Erfassen Stichwörter“

Die Auswahl „Stichworte“ öffnet einen Dialog, in dem Stichworte angegeben werden können, mit denen ein Baustein in der Datenbank gefunden werden kann. In Abbildung 99 ist neben dem Erfassen der Stichworte auch der Suchdialog dargestellt. Über den Suchdialog kann ein Modell bzw. ein Baustein durch Angabe eines Stichworts ermittelt werden.

Versionen

Die unterschiedlichen Versionen eines Bausteins können durch ein Formular geändert werden, das in Abbildung 100 dargestellt ist. Der Zeitraum, in dem eine neue Version eines Bausteins erstellt worden ist, wird durch die Felder „Beginn“ und „Ende“ angegeben. Die Angabe

der Simple Version ist erforderlich, da es bei Releasewechselln zu Änderungen kommen kann, die den Einsatz eines Bausteins in höheren Entwicklungsversionen von eM-Plant einschränken. Neue Versionen können dokumentiert bzw. bestehende Dokumentationen aktualisiert werden. Die in der ausgewählten Version enthaltenen Methoden werden in einem Dialog angezeigt.



Version	Unterversion
6	9
6	10
6	11
6	12
6	13
6	14
6	15
6	16
6	17
6	18
7	1
8	1
8	2
8	3
8	4
8	5

Abbildung 100 - Bausteinverwaltung: Dialog „Erfassen Versionen“

Erfassen von Modellen

Auch Modelle können erfasst und verwaltet werden. Dabei werden das Modell, der Modellverantwortliche, der aktuelle Modellstand etc. dokumentiert. Abbildung 6 zeigt das entsprechende Formular. Zur Wartung der Modelle werden zusätzlich die Versionsnummer eines Bausteins angegeben, der im Modell eingesetzt ist. Zusätzlich kann freier Text eingegeben bzw. angezeigt werden.

Modell- und Bausteinerweiterungen

Soll ein Modell oder ein Baustein um Funktionalität erweitert werden, so können die erforderlichen Arbeiten erfasst werden. Hierzu ist die Funktion „ToDo“ auszuwählen. Im eingeblendeten Fenster können dann kurze Informationen zu den geplanten Änderungen eingegeben werden. Alle erfassten Aufgaben können im Dialog „Anzeige offener Arbeitspunkte“ angezeigt werden (Abbildung 102). Ist die Liste der offenen Arbeitspunkte für ein Modell bzw. Baustein zu umfangreich, kann nach dem gewünschter Baustein bzw. Modell gefiltert werden. Ob sich ein offener Arbeitspunkt auf einen Baustein oder auf ein Modell bezieht, ist aus dem Typ ersichtlich (Bausteine erhalten den Typ „B“, Modelle den Typ „M“).

Erfassen Modell

Bezeichnung: ?

Verantwortlich:

Position:

Beginn: ▼

Letzter Stand: ▼

Bausteine	Version	UVersion
Terminierung	1	5
Data	4	5
Datenbaustein	6	3
Pausen	10	2
ObjektBackup	5	1
Objektauswahl	3	3
TabellenFilter	2	4
Basistoolset	3	5
Fortschritt	2	1
search_for_objects	2	2
search_in_Methods	1	2
Modellverwaltung	1	2
ObjektTransfer	1	1
Pause	8	6
Gantt	6	8

Dokumentation

- Terminieren einer Planungsvariante
- Visualisierung der Terminlage mit Gantt-Charts
- Zurückschreiben der Termine in die Datenbank(en)

Anlegen Bausteine Stichworte ToDo Abbrechen

Abbildung 101 - Bausteinverwaltung: Dialog „Erfassen Modell“

Anzeigen offene Arbeitspunkte

lfd. Nummer	Baustein	Typ	Beschreibung	bearbeitet
2	Leitrechner	M	Zuordnung zu einem Frame entspricht der Zuordnung zu allen auftragaufnehmenden Instanzen im entsprechenden Frame.	<input type="checkbox"/>
3	Leitrechner	M	Pausenmeldung usw. an Prozesssteuerung, die den gerade bearbeiteten Auftrag verwaltet.	<input type="checkbox"/>
1	Objektauswahl	B	Evtl. Erweiterung der Funktion: Auswahl von Attributen (z.B. Methoden als Attribute von Objekten)	<input type="checkbox"/>
1	ObjektBackup	B	Hatten Tabellen des alten Bausteins das Attribut "Reproduktion", so sollte auch dieses Attribut selbst am neuen Baustein reproduziert werden	<input checked="" type="checkbox"/>
2	ObjektBackup	B	Auswahl und übersichtliche Darstellung der Objekte (Tabellen), an der die Reproduktion vorgenommen werden soll	<input type="checkbox"/>
3	ObjektBackup	B	Ausführlichere Fortschrittsanzeige (Backup nimmt u.U. viel Zeit in Anspruch)	<input type="checkbox"/>

Arbeitsplanmanager
Basistoolset
BenutzerKontextMenü
BestandGantt
Data
Gantt
KL_SQL
Koordinaten
Leitrechner
ObjektBackup
Objektauswahl
Pausen
Ressourcenfinder
Search_in_methods
Terminierung
Verfügbarkeit

Alle Punkte anzeigen Abbrechen

Abbildung 102 - Bausteinverwaltung: Dialog „Anzeigen offener Arbeitspunkte“

2.6 Sonstige Arbeitspunkte

Experimentverwaltung

Im Projekt zeigte sich das die Verwaltung von Experimenten im Gegensatz zur klassischen Simulation in der Serienfertigung, bei der Simulation in der Einzelfertigung nicht von entscheidender Bedeutung ist. Zwar werden auch einige Simulationsexperimente durchgeführt, die Auswertung und das Anstoßen von neuen Experimenten erfolgt manuell durch den Anwender. Die Variation der Simulationsparameter erfolgt mit dem SDM Tool. Die Verwaltung der Simulationsdaten erfolgt in Varianten. Durch diese Arbeitsweise wurde bei der Entwicklung des Datenmodells die Variantenverwaltung besonders komfortable gestaltet. Eine zusätzliche Experimenten Verwaltung auf der Simulationsebene war im Projekt nicht erforderlich.

Wissensmodellierung (Abbildung von „Know How“)

Durch die Entwicklung eines integrierten Datenmodells mussten für die Modellierung von Wissen („Fertigungs- Know How“) keine speziellen Werkzeuge entwickelt werden. Die Modellierung von „Fertigungs- Know How“ erfolgt im Datenmodell in den jeweiligen Datenkomplexen. Im Simulationsmodell wird das Know How von den entsprechenden Komponenten (Leitrechner etc). berücksichtigt. Mit dem SDM Tool können diese Daten jederzeit gepflegt werden. Das „Fertigungs Know How“ kann also ständig gepflegt und aktualisiert werden. Neue Strategien können vom Simulationsexperten in die entsprechenden Baustein integriert werden. Beide Vorgehensweisen wurden im Projekt ständig verwendet.

3 Zusammenfassung

Im Projekt „Simulationstoolset für die Werftindustrie“ wurden verschiedenste Aspekte der Materialflusssimulation einer Kompaktwerft untersucht. Dabei wurden sowohl neue wissenschaftliche Erkenntnisse gewonnen, als auch erste Einsatzerfahrungen gesammelt und Anwendungen realisiert. Nicht alle Ergebnisse konnten im Rahmen dieses Berichtes in gebührender Form vorgestellt werden. Die Meyer Werft ist, wie auch bereits während der Laufzeit des Projekts, zur Zusammenarbeit bei der Weiterentwicklung von Simulationstechniken und deren Einsatz bereit.

Durch das „Simulationstoolset“ konnte der Modellierungsaufwand bei der Modellerstellung wesentlich verringert werden. Planungs- Simulationslösungen können somit schneller dem Endanwender zur Verfügung gestellt werden. Dies gilt sowohl für den Bereich der Investitionsplanung als auch der Fertigungsplanung. Basis für alle Simulationsanwendungen der Meyer Werft ist das entwickelte Simulationstoolset. Das Simulationstoolset und die Simulationsanwendungen werden vom Simulationsexperten auch in Zukunft weiterentwickelt.

Zum Projekterfolg hat die Entwicklung des integrierten Datenmodells und des Datenmanagementtools wesentlich beigetragen. Für den Anwender wurde für alle Simulationsanwendungen eine einheitliche Simulationsumgebung geschaffen. Das integrierte Datenmodell erfordert eine sehr strukturierte Arbeitsweise mit der Simulation. Gleichzeitig bietet das Datenmodell eine sehr hohe Flexibilität, die es dem Anwender ermöglicht, seine Prozesse innerhalb der Simulation abzubilden. Durch das Datenmodell wird eine sehr hohe Transparenz im Planungsprozess für alle Simulationsanwender erzielt. Gleichzeitig können alle Simulationsanwender mit den gleichen Daten arbeiten.

Durch den Fokus auf den Endanwender konnte im Projekt eine Simulationsumgebung erstellt werden, die es auch dem Simulationslaien ermöglicht die Simulation sinnvoll zu nutzen. Bei der Entwicklung des Datenmanagementtools wurden besonders die Anforderungen an Interaktivität und anwenderfreundliche Oberflächen berücksichtigt.

Mit der Einbindung von simulationsfremden Systemen wie z.B. CAD Datentransfer, Schnittstellen zur Ablaufplanung, Optimierung etc. konnte aufgezeigt werden, dass mit einer klaren Strategie die Simulation vollständig in den Planungsprozess einer Werft eingebunden werden kann. Die Integration der Simulation in die EDV Umgebung der Meyer Werft wird auch in Zukunft weiter vorangetrieben.

Die im Projekt erreichten Ergebnisse sind in vielen Punkten sehr innovativ. Zahlreiche Diskussionen mit anderen Werften haben den innovativen Charakter des Projektes bestätigt. Das Projekt hat durch die ganzheitliche Betrachtungsweise der Simulation in vielen Bereichen dort weitergearbeitet, wo andere bereits aufgehört haben. Die Simulation wird den Planungsprozess der Meyer Werft als ein wichtiges Werkzeug stark unterstützen. Die bisher erstellten Pilotanwendungen werden in Zukunft weiterentwickelt und weiter in die Planungsumgebung der Meyer Werft integriert. Auf Basis der bisher erstellten Anwendungen zeichnen sich noch viel Einsatzmöglichkeiten für Simulationsanwendungen ab.

Abbildungsverzeichnis

Abbildung 1 - Projektziele/Aufgabenkomplexe.....	9
Abbildung 2 - Software-Auswahlprozess: Benchmark-Modell.....	13
Abbildung 3 - Datenmodell.....	19
Abbildung 4 - Bauteilkoordinatenverwaltung.....	20
Abbildung 5 - Schiffskoordinaten.....	20
Abbildung 6 - Umsetzung Datenstruktur.....	22
Abbildung 7 - MethodManager: Eingabemaske.....	25
Abbildung 8 - Objektaufbau: Beispiel Pausenbaustein.....	26
Abbildung 9 - Objektgestaltung: Formatvorlage für Anwenderbausteine.....	27
Abbildung 10 - Basistoolset: Aufbauhierarchie.....	28
Abbildung 11 - Dokumentationsbaustein: Anlegen einer neuen Dokumentation.....	30
Abbildung 12 - Dokumentationsbaustein: Dokumentation bearbeiten.....	31
Abbildung 13 - Dokumentationsbaustein: weitere Funktionen.....	33
Abbildung 14 - Modellverwaltung: Dialoge und Funktionen.....	34
Abbildung 15 - Fortschritt: Beispiel Fortschrittmeldung.....	36
Abbildung 16 - Objektauswahl: Mehrfachauswahl mit TreeView und ListView.....	37
Abbildung 17 - Objektauswahl: Eingabeaufforderung mit promptList.....	37
Abbildung 18 - Objektauswahl: Beispiel Methode ObjekteFilter.....	38
Abbildung 19 - ObjektBackup: Anwendungsbeispiel.....	39
Abbildung 20 - ObjektTransfer.....	40
Abbildung 21 - searchInMethods: Benutzeroberfläche und Ergebnistabelle.....	41
Abbildung 22 - search_for_objects: Benutzeroberfläche und Ergebnistabelle.....	42
Abbildung 23 - search_in_tables: Benutzeroberfläche und Ergebnistabelle.....	43
Abbildung 24 - Datenbaustein: Struktur.....	47
Abbildung 25 - Datentabelle: Instanz Datentabelle Rsc_EIG_Orientieru.....	48
Abbildung 26 - Datentabelle: Instanz Informationstabelle Rsc_EIG_Orientieru.....	48
Abbildung 27 - Datenbaustein: GUI.....	49
Abbildung 28 - Datenbaustein: Arbeitsplanstruktur.....	50
Abbildung 29 - Datenbaustein: Prozessstruktur.....	50
Abbildung 30 - Datenbaustein: Prozesszeitermittlung.....	50
Abbildung 31 - Datenbaustein: Ressourcenstruktur.....	51
Abbildung 32 - Datenbaustein: Beispiel Get_Tableinfo, Rückgabetabelle für Tabelle Rsc_EIG_Qualifik.....	52

Abbildung 33 - Datenbaustein: Beispiel Get_Select / Get_Data_Select, Auswahl- und Rückgabetabelle für Zugriff auf Tabelle Rsc_EIG_Qualifik.....	52
Abbildung 34 - Data-Baustein: Aufbau	54
Abbildung 35 - SQL-Baustein: Schema der Tabelle „Datenstruktur“	57
Abbildung 36 - SQL-Baustein: Aufbau einer Informationstabelle	58
Abbildung 37 - SQL-Baustein: Inhalt einer Datentabelle mit grau hinterlegten Schlüsselfeldern	58
Abbildung 38 - SQL-Baustein: Zusammenhang Tabellenbeschreibung Access / Informationstabelle eM-Plant.....	59
Abbildung 39 - SQL-Baustein: Netzwerk der SQL Schnittstelle.....	60
Abbildung 40 - Manager auf Ressourcennetzwerk- und Ressourcenebene: Bauteilrepräsentant mit Leitrechnerfunktionen.....	64
Abbildung 41 - Transportmanager: Funktionsübersicht	66
Abbildung 42 - Get_Transportwege Rückgabetabelle	69
Abbildung 43 - Get_Transportwege Untertabelle Transportwege.....	70
Abbildung 44 - Get_Rsc_Belastung Rückgabetabelle	70
Abbildung 45 - Get_Pakete_Belastung Rückgabetabelle	71
Abbildung 46 - Ressourcenklassen: Typisierung	73
Abbildung 47 - Paneelstraße: Klassenbibliothek	76
Abbildung 48 - Paneelstraße: Layout des Ressourcennetzwerks.....	76
Abbildung 49 - Paneelstraße: Erweiterung	77
Abbildung 50 - Platz	78
Abbildung 51 - Platz: Platzkopplung	79
Abbildung 52 - Platz: Einsetzen des selben Bauteils auf Plätzen unterschiedlicher Orientierung	79
Abbildung 53 - Fahrspur: Fahrspurlayout.....	80
Abbildung 54 - Modelle erstellen: Beispiel	89
Abbildung 55 - Raumbaustein: Übersicht der Komponenten	92
Abbildung 56 - Pausenbaustein: Einsatz und Dialoge	94
Abbildung 57 - Gantt: Hauptdialog.....	97
Abbildung 58 - Gantt: Beispiele verschiedener Darstellungsmodi	98
Abbildung 59 - Gantt: Darstellungsmodus konfigurieren.....	99
Abbildung 60 - Gantt: Auswahl Zusatzinformationen.....	100
Abbildung 61 - Gantt: Definition von Filtern	100
Abbildung 62 - BestandGantt / BestandDiagramm: Anwendungsbeispiel.....	101
Abbildung 63 - Eventschnittstelle	102

Abbildung 64 - Funktion Eventschnittstelle.....	102
Abbildung 65 - Oberflächen: ActiveX-Steuerungen (Auswahl).....	105
Abbildung 66 - Datenaustausch über Datenpakete	106
Abbildung 67 - Datenpaket	106
Abbildung 68 - Kommunikationskomponente	107
Abbildung 69 - Meldungsverarbeitung durch den Kommunikationsbaustein	108
Abbildung 70 - Kommunikation: Datenaustausch über Datenpakete	109
Abbildung 71 - Kommunikation: Datenpaket.....	109
Abbildung 72 - Kommunikation: Kommunikationskomponente.....	110
Abbildung 73 - Meldungsverarbeitung durch den Kommunikationsbaustein	111
Abbildung 74 - SDM Tool: Komplex „Bauplan“	114
Abbildung 75 - SDM-Tool: Komplex Operationsplan	115
Abbildung 76 - SDM Tool: Komplex „Operationsplan“	116
Abbildung 77 - SDM Tool: Interaktives Gantt- und Kapazitätsdiagramm	117
Abbildung 78 - SDM Tool: Interaktive Plantafel.....	118
Abbildung 79 - Modellverwaltung.....	119
Abbildung 80 Virtuelle Werft: Workflow für Simulationsstudien	121
Abbildung 81 - Sim_Schedule: Hauptdialog	122
Abbildung 82 - Sim_Schedule: Dialog Rahmentermine.....	123
Abbildung 83 - Sim_Schedule: Festlegen der auszuführenden Funktionen	124
Abbildung 84 - Sim_Schedule: Abgleich.....	125
Abbildung 85 – Sim-Space: Übersicht.....	128
Abbildung 86 - Grobsimulation: erstes Modell.....	129
Abbildung 87 - Simulation Paneelstraße: 2D-Animation	131
Abbildung 88 - Simulation Paneelstraße: Eingangsdaten, Parametrierung	133
Abbildung 89 - Simulation Paneelstraße: Auswertung.....	134
Abbildung 90 - CAD-Datentransfer: CATIA Modell / Produktmodell	136
Abbildung 91 - Datengenerierung: Interaktive Modellierung von Arbeitsschritten	137
Abbildung 92 - Datengenerierung: Generator.....	138
Abbildung 93 - Optimierung: Aufbau des Netzwerks.....	140
Abbildung 94 – Optimierung: Fitness.....	141
Abbildung 95 - Optimierung: Entwicklung der Fitnesswerte	146
Abbildung 96 - Optimierung: Ergebnisse Simulation.....	147
Abbildung 97 - Optimierung: Ergebnisse Optimierung.....	147



Abbildung 98 - Bausteinverwaltung: Dialog „Erfassen Bausteine“	150
Abbildung 99 - Bausteinverwaltung: Dialoge „Suchen Komponenten“ (links) und „Erfassen Stichwörter“	150
Abbildung 100 - Bausteinverwaltung: Dialog „Erfassen Versionen“	151
Abbildung 101 - Bausteinverwaltung: Dialog „Erfassen Modell“	152
Abbildung 102 - Bausteinverwaltung: Dialog „Anzeigen offener Arbeitspunkte“	152

Tabellenverzeichnis

Tabelle 1 - Kriterien und Bewertung des eM-Plant Benchmark-Modells.....	13
Tabelle 2 - Dokumentationsbaustein: allgemeine bausteinbezogene Informationen.....	30
Tabelle 3 - Dokumentationsbaustein: allgemeine versionenbezogene Informationen.....	31
Tabelle 4 - Tabellenfilter: Operatoren für Methode <i>findeDatensätze</i>	45
Tabelle 5 - Ressourcenklassen: Öffentliche Methoden der Ressourcenklassen	81
Tabelle 6 - Meldungen Maschine_Beweglich: Rückgabe Status_Ermitteln.....	84
Tabelle 7 - Pausenbaustein: Öffentliche Methoden	95
Tabelle 8 - Gantt: voreingestellte Darstellungsmodi.....	97
Tabelle 9 - Meldungen Kommunikationsbaustein: Meldungen Link	108
Tabelle 10 - Meldungen Kommunikationsbaustein: Meldungen Unlink.....	109
Tabelle 11 - Meldungen Kommunikationsbaustein: Meldungen Link	111
Tabelle 12 - Meldungen Kommunikationsbaustein: Meldungen Unlink.....	111
Tabelle 13 - Optimierung: Kriterien und Vergleich von Optimierung und Simulation.....	148

Literaturverzeichnis

- [1] Roland, F.: Strategiepapier für F&E - Vorhaben "Blockbau und Vorausrüstung", Meyer Werft, Papenburg Dezember 1995
- [2] VDI-Richtlinie 3633 (Entwurf): Simulation von Logistik-, Materialfluss- und Produktionssystemen. Blatt 1-4. Verein Deutscher Ingenieure, Düsseldorf 1996
- [3] LANNER GROUP: Handbücher zum Simulationssystem WITNESS.
- [4] AESOP GmbH: Produktinformationen und Handbücher zum Simulationssystem Simple++
- [5] Aga, S.; Hatling, J.F.: Simulation - A Powerful Tool for the Shipbuilding Industry. 8th International Conference on Computer Applications in Shipbuilding (ICCAS 1994). Bremen 1994. Proceedings Volume 1, S. 8:103 ff.
- [6] Kattan, R.M.; Weissshuhn, A.; Lightfoot, J.: Simulation in Shipbuilding. Vortrag, gehalten vor der Royal Society of Naval Architects 1996. Keine weiteren Angaben.
- [7] Projektpräsentation "Simulationsbasierte Logistik- und Produktionsstrategien im Schiffbau", Workshop Peenewerft - Meyer Werft, Wolgast 20.01.1998
- [8] Kuhlmann, T.; Marciniak, Z.; Maßow, C.: Integrated Coordination Modules for the Shipbuilding Industry. 8th International Conference on Computer Applications in Shipbuilding (ICCAS 1994). Bremen 1994. Proceedings Volume 2, S. 12:39 ff.
- [9] Alkaner, S.; Ölcer, I.A.: Ship Production Scenario Analysis using Multiple Operations Research Techniques. 8th International Conference on Computer Applications in Shipbuilding (ICCAS 1994). Bremen 1994. Proceedings Volume 1, S. 8:85 ff.
- [10] Ishiyama, M.; Abe, A.: Study on Quantitative Evaluation of Workability in Shipbuilding. 8th International Conference on Computer Applications in Shipbuilding (ICCAS 1994). Bremen 1994. Proceedings Volume 1, S. 8:69 ff.
- [11] Alfeld, L.E. u.a.: The Virtual Shipyard: A Simulation Model for the Shipbuilding Process. Journal of Ship Production 14(1998) S.33-40
- [12] Johansson, K.; 9th International Conference on Computer Applications in Shipbuilding (ICCAS 1997). Yokohama, Japan 1997. Proceedings Volume 1+2
- [13] Johansson, K.; 10th International Conference on Computer Applications in Shipbuilding (ICCAS 1999). Boston, Japan 1999. Proceedings Volume 1+2
- [14] Kosturiak, J.; Gregor M.; Simulation von Produktionssystemen. 1995 Springer Verlag/Wien
- [15] Kuhn, A.; Rabe M.; Simulation in Produktion und Logistik. 1998 Springer Verlag
- [16] „Konzeption und Realisierung einer grafischen Schnittstelle für die Simulation von Werftabläufen“. Diplomarbeit Wilfried Kloppenburg, Papenburg 2001.
- [17] „Simulation als Betriebliche Entscheidungshilfe in der Schiffbauindustrie“. Diplomarbeit Hans-Jürgen Lüken, Papenburg 1999.

Glossar

Verzeichnis verwendeter Akronyme

Akronym	Bedeutung	Erklärung
BE	Bewegliches Element	Bausteine (Fördergut, Förderhilfsmittel, Fahrzeug) in eM-Plant, die während eines Simulationslaufs auf den Materialflussbausteinen (Ressourcen) durch das Modell bewegt werden.
eM-Plant		bei JLM verwendete Basissoftware für Materialflusssimulation (Umbenennung)
GUI	Graphical User Interface	Graphische Benutzeroberfläche
JLM	Jos.L.Meyer GmbH	Meyer Werft
KTR	Kostenträger	Schiff im Auftragsbestand der Werft
SDM	Simulationsdatenmanagement	Im Projekt entwickeltes Programm mit graphischer Oberfläche zum Generieren und Bearbeiten von Simulationsdaten (VB)
Simple++		bei JLM verwendete Basissoftware für Materialflusssimulation (ab 1998)
SimTalk		Programmiersprache, mit der in eM-Plant benutzerdefinierte Methoden programmiert werden
SQL	Structured Query Language	Datenbankabfragesprache
SQL-Server		relationales Datenbankmanagementsystem
VB	Visual Basic	Programmiersprache
Witness		bei JLM verwendete Basissoftware für Materialflusssimulation bis 1998

Verzeichnis verwendeter Fachbegriffe

Begriff	Erklärung
Methode, benutzerdefinierte	Basisbaustein in eM-Plant, in dem in der Programmiersprache SimTalk benutzerdefinierte Steuerungen programmiert werden können. Deren Ausführung kann dann z.B. zur Simulationszeit durch Ereignisse im Materialfluss oder von anderen Methoden ausgelöst werden.
Netzwerk	Basisbaustein in eM-Plant, der andere Basisbausteine und auch Netzwerke aufnehmen kann. Netzwerke werden zur Erstellung hierarchisch strukturierter und logisch gegliederter Simulationsmodelle verwendet.