

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Aufbau der Arbeit	3
1.3	Anwendungsbeispiele	4
1.3.1	Linearer Stellantrieb	4
1.3.2	Edutruck	5
1.3.3	Servicezubringer	7
2	Entwurf mechatronischer Systeme	9
2.1	Kennzeichen mechatronischer Systeme	9
2.2	Einordnung des Entwurfs in den Produktlebenszyklus	10
2.3	Stand der Technik beim Entwurf mechatronischer Systeme	12
2.4	Analyse existierender Softwaresysteme	14
2.4.1	Numerische Berechnungsprogramme	14
2.4.2	Computeralgebra-Systeme	17
2.4.3	Berechnungsprogramme für mechanische Aufgabenstellungen	18
2.4.4	CAMeL-View – eine Entwicklungsumgebung für mechatronische Systeme	19
3	Strukturierung des Entwurfsprozesses	21
3.1	Phasen im Entwurfsprozess	21
3.2	Kennzeichen der Top-Down- und Bottom-Up-Entwurfsmethode	23
3.3	Arbeitsprozesse beim Entwurf mechatronischer Systeme	23
3.4	Der modellbasierte ganzheitliche Entwurf	25
3.5	Drei mögliche Sichtweisen auf ein mechatronisches System	26
3.5.1	Funktionensicht	26
3.5.2	Hierarchiesicht	28
3.5.3	Modellsicht	30
3.6	Zusammenfassende Kennzeichnung des Entwurfsverfahrens	32
4	CAMeL-View – Konzept und Implementierung	33
4.1	Kennzeichen objektorientierter Programmierung	37
4.2	Überblick über objektorientierte Programmiersprachen	38
4.3	Der Komponentenbegriff in der Softwareentwicklung	40
4.4	Mechanismen zur Kommunikation zwischen Komponenten	42
4.4.1	DDE und Pipe-Mechanismus	43
4.4.2	COM	43
4.4.3	.NET	44
4.4.4	CORBA	45
4.4.5	J2EE	46
4.4.6	Gegenüberstellung der Kommunikationstechniken	48
4.5	Externe Softwarepakete in CAMeL-View	48
4.5.1	Plotter	48
4.5.2	Scilab	50

4.5.3	MuPAD	51
4.5.4	Benutzerschnittstellen zu externen Tools	51
4.5.5	Weitere externe Tools in CAMEL-View	52

5 Beschreibungsmform „Workflow“ **55**

5.1	Überblick zur Entwicklung von Workflows	55
5.1.1	Begriffsbestimmung „Workflow“	56
5.1.2	Begriffsbestimmung „Workflow-Management-System“	57
5.1.3	Phasen eines Workflows	57
5.2	Einsatzgebiete heutiger Workflow-Management-Systeme	58
5.3	Realisierung von Ablaufsteuerungen in CAMEL	58
5.3.1	Skriptbasierte Ablaufsteuerung – Experimentbeschreibungssprache	59
5.3.2	Ablaufsteuerung durch eine anwendungsspezifische Benutzeroberfläche	62
5.4	Anforderungen an Workflows beim Entwurf mechatronischer Systeme	64

6 Modellierung von Workflows **67**

6.1	Kennzeichnung der notwendigen Workflow-Module	67
6.1.1	Module zur Informationsbeschaffung und -verwaltung	67
6.1.2	Module für Analyse- und Synthesaufgaben	68
6.1.3	Module zur Steuerung der Abarbeitung	69
6.1.4	Module zur Workflowsteuerung	70
6.2	Zusammenführung der Einzelkomponenten	70
6.2.1	Initialisierung externer Softwarekomponenten	71
6.2.2	Prozess zur Abarbeitung einer Aktivität	72
6.2.3	Prozess zur Entscheidungsfindung und Verzweigung der Abarbeitungsstränge	73
6.2.4	Visualisierung von Analyseergebnissen im Workflow	75
6.2.5	Verkopplung der Module zum Workflow	75
6.3	Workflow-Beschreibungselemente	76
6.3.1	Grundaufbau der Beschreibungselemente	78
6.3.1.1	Klassendefinition	80
6.3.1.2	Schnittstellendefinition	81
6.3.1.3	Implementierung	81
6.3.2	Klassen zur Workflowsteuerung	82
6.3.3	Klassen zur Initialisierung externer Werkzeuge	83
6.3.3.1	Skriptgesteuerte Programmpakete	83
6.3.3.2	Komponenten mit Konfigurationsattributen	84
6.3.4	Klassen zur Beschreibung von Abarbeitungsprozessen	85
6.3.4.1	Klasse zur Abbildung eines Smalltalk-Skripts	85
6.3.4.2	Klasse zur Abbildung eines skriptbasierten Prozessaufrufs	86
6.3.4.3	Klasse zur Abbildung eines Prozesses mit Konfigurationsattributen	87
6.3.5	Klassen zur Abbildung von Entscheidungen	88
6.3.5.1	Automatisierte Entscheidung	88
6.3.5.2	Benutzergeführte Entscheidung	89
6.3.6	Klassen zur Abbildung von Verzweigungen im Workflow	91
6.3.7	Klassen zur Zusammenführung alternativer Abarbeitungsstränge	93
6.3.8	Klassen zur Abbildung von Hierarchien	94
6.3.9	Koppelemente	96
6.4	Ausgewählte Beispiele	96

6.4.1	Sequenzieller Workflow	96
6.4.2	Verzweigter Workflow	97
6.4.3	Schleifenstrukturen	98
6.5	Modellverwaltung in CAMEL-View	98
6.5.1	Verwaltung der Workflow-Modelle	98
6.5.2	Visualisierung der Workflow-Modelle	99
6.6	Graphenbasierte Repräsentation der Workflow-Modelle	99
6.6.1	Graphen und Algorithmen	100
6.6.2	Abbildung modular-hierarchischer Systeme	101
6.6.3	Analyse der Workflow-Struktur	101
6.6.3.1	Prüfung der strukturellen Konsistenz	102
6.6.3.2	Prüfung der strukturellen Korrektheit	103
7	Abarbeitung von Workflows	109
7.1	Abarbeitungskonzept	109
7.1.1	Lauffähige Objekte zur Abarbeitung einer Aktivität	110
7.1.2	Ergebnisorientierte Abarbeitung einer Aktivität	111
7.1.3	Datenaustausch zwischen den Aktivitäten	115
7.1.4	Berücksichtigung äußerer Ereignisse bei der Abarbeitung eines Workflows	116
7.2	Generierung des lauffähigen Skripts	117
7.2.1	Instanziierung der lauffähigen Objekte	119
7.2.2	Instanziierung der Verbindungen zwischen den lauffähigen Objekten	119
7.3	Aufbau einer Laufzeitumgebung	120
7.3.1	Steuerung der Laufzeitumgebung durch den Anwender	121
7.3.2	Überwachung der Aktivitäten	122
7.3.3	Debuggen eines Workflows	123
7.3.4	Error-Handling beim Ausführen eines Workflows	124
7.4	Rekonfiguration eines Workflows zur Laufzeit	125
7.4.1	Phasen im Rekonfigurationsvorgang	125
7.4.2	Aufbau eines Rekonfigurationsskripts	127
8	Realisierung des Workflow-Systems in CAMEL-View	129
8.1	Integration des Workflow-Systems in CAMEL-View	129
8.1.1	Erweiterung von CAMEL-View um einen Workflowmodellierer	130
8.1.2	Integration der Workflow-Laufzeitumgebung in CAMEL-View	132
8.2	Anwendungsbeispiele für Workflows	134
8.2.1	Workflowgestützte Auslegung eines Zustandsreglers	134
8.2.1.1	Auslesen des Streckenmodells aus der OdssDatabase	137
8.2.1.2	Linearisierung des Streckenmodells	138
8.2.1.3	Überprüfung der Strecke auf Steuerbarkeit	139
8.2.1.4	Verzweigung des Workflows in Abhängigkeit von der Steuerbarkeit	140
8.2.1.5	Auslegung eines Reglers zur Stabilisierung des Systems	142
8.2.1.6	Speicherung der Rückfühmatrix in der OdssDatabase	143
8.2.1.7	Zusammenführung der Basissysteme zu einem Workflow	144
8.2.2	Workflowgestützte Auslegung einer Fahrdynamikregelung	145
8.2.3	Workflowgestützte Systemoptimierung	149
8.3	Übertragbarkeit des Workflowkonzepts auf weitere Anwendungsfälle	151
8.3.1	Prüfstandsautomatisierung mit Hilfe von Workflows	151

8.3.2	Softwareentwicklung mit Hilfe von Workflows	151
9	Zusammenfassung und Ausblick	153
A	Grammatik	157
A.1	Metaklassen zur Workflow-Steuerung	157
A.1.1	StartOwss	158
A.1.2	FinishOwss	158
A.2	Metaklassen zur Initialisierung von Softwarewerkzeugen	159
A.2.1	MupadInit	159
A.2.2	ScilabInit	160
A.2.3	PlotterInit	161
A.3	Metaklassen zur Beschreibung von Abarbeitungsprozessen	161
A.3.1	SmalltalkProcess	162
A.3.2	MupadProcess	163
A.3.3	ScilabProcess	164
A.3.4	PlotterProcess	165
A.3.5	UserDecisionProcess	166
A.3.6	CodedDecisionProcess	167
A.3.7	SplitProcess	168
A.4	Metaklassen zur Beschreibung von Hierarchien	169
A.4.1	WorkflowHcsOwss	169
A.5	Metaklassen zur Parametrisierung des Workflows	171
A.5.1	ParametrizationOwss	171
B	Beispiel: Modellierung und Reglerauslegung des linearen Stellantriebs ..	173
B.1	Systembeschreibung	173
B.2	Struktur des mechatronischen Systems	174
B.3	Lösungselemente beim linearen Stellantrieb in reduzierter Darstellungsform	174
B.3.1	Modell des Gleichstrommotors	175
B.3.2	Verhalten des Gleichstrommotors	177
B.3.3	Modellierung des Schubkurbeltriebs	180
B.3.3.1	Übersetzung von ϕ nach δX	180
B.3.3.1	Übersetzung von F_x nach M_A	181
B.3.4	Modellierung des Motorgetriebes	182
B.3.5	Modellierung der Hebelarmverstellung	183
B.4	Zusammenführung der einzelnen Komponenten zum Gesamtmodell	184
B.4.1	Linearisierung des Gesamtsystems im Betriebspunkt	189
C	Beispiel: Modellierung und Reglerauslegung beim Edutruck	191
C.1	Edutruck Gesamtmodell	191
C.2	Abbildung der Querdynamik im Einspurmodell	192
C.3	Das linearisierte Einspurmodell	194
C.4	Auslegung der Fahrdynamikregelung	196
	Literatur	201