

Konzeption einer Architektur für Optimierungssoftware zur Integration in betriebliche Anwendungssysteme

Dissertation
zur Erlangung des akademischen Grades
Doktor der Wirtschaftswissenschaft
vorgelegt von

Dipl. Kfm. Peter Ingerfeld

Freie Universität Berlin
Fachbereich Wirtschaftswissenschaft
Betreuer: Prof. Dr. Uwe Suhl
Koreferent: Prof. Dr. Achim Koberstein

Tag der Disputation: 07.07.2008

Inhaltsverzeichnis

0	Abstract.....	5
1	Einleitung und Zielsetzung der Arbeit	7
2	Einführung	9
2.1	Anwendungsgebiete linearer und gemischt-ganzzahliger Modelle.....	9
2.2	Architektur Entscheidungsunterstützender Systeme (EUS).....	12
2.2.1	Entscheidungsunterstützende Systeme	12
2.2.2	Integration von Optimierungsfunktionalität in EUS	15
2.2.3	Umfeldfunktionalitäten eines Optimierungssystems.....	18
2.2.4	Schnittstellenproblematik	19
2.3	Ziele der Arbeit.....	26
2.4	Abgrenzung	27
2.5	Angewendete Forschungsmethoden	29
2.6	Aufbau der Arbeit.....	31
3	State-of-the-art von Solver- und Modellierungssystemschnittstellen	33
3.1	Prozedurale Optimierungssystemschnittstellen.....	34
3.1.1	Untersuchungsrahmen	34
3.1.2	Funktionalitäten	37
3.1.2.1	Umgebungs- und Modellhandling	37
3.1.2.2	File I/O.....	38
3.1.2.3	Modellaufbau und -modifikation.....	40
3.1.2.4	Modellabfrage.....	41
3.1.2.5	Lösungsabfrage und -analyse	42
3.1.2.6	Lösungsprozesssteuerung	43
3.1.2.7	Parameterhandling	43
3.1.2.8	Callbacks	44
3.1.2.9	Fehlermeldungen und sonstige Funktionen.....	45
3.2	Objektorientierte Optimierungssystemschnittstellen.....	45
3.2.1	Klassenbibliotheken	46
3.2.1.1	BCL	46
3.2.1.2	ILOG Concert Technology Library.....	49
3.2.1.3	Vergleich BCL und Concert Technology.....	54
3.2.2	Komponentenbibliotheken	55
3.2.2.1	XPRESS-MP .NET.....	55
3.2.2.2	Concert Komponentenbibliothek für .NET	56
3.2.2.3	OptiMax 2000.....	58
3.3	Integrierte Modellierungssysteme	58
3.3.1	ILOG OPL Development Studio	60
3.3.2	XPRESS IVE.....	60
3.3.3	AIMMS.....	61
3.4	Modellierungssprachen.....	63
3.4.1	Gemeinsame Sprachelemente.....	65
3.4.2	Besonderheiten der untersuchten Modellierungssprachen	77
3.4.3	Vergleich ausgewählter Eigenschaften.....	84
4	Softwarearchitektonische Determinanten.....	85
4.1	Prozedurale und objektorientierte Programmierparadigmen.....	85
4.2	Komponentenmodelle.....	87
4.2.1	Softwarekomponenten	87
4.2.2	Komponentenframeworks	89
4.2.3	Standards	91

4.2.3.1	CORBA.....	92
4.2.3.2	Java-Komponenten.....	93
4.2.3.3	COM und .NET.....	96
4.2.4	Komponentenmärkte.....	100
4.3	Bibliotheks- und Schnittstellendesigns unter Windows.....	102
4.3.1	Statische und dynamische Bibliotheken.....	102
4.3.2	Klassenbibliotheken.....	103
4.3.3	Component Object Model.....	104
4.3.3.1	Das Basisinterface IUnknown.....	104
4.3.3.2	Class Factories.....	106
4.3.3.3	Die Schnittstelle IDispatch.....	107
4.3.3.4	Lokale und Remote-Ausführung.....	108
4.3.3.5	Objektorientierte Charakteristika in COM.....	109
4.3.3.6	Property Pages.....	111
4.3.3.7	Bibliotheken zur COM-Programmierung.....	112
4.3.4	.NET als neues Komponenten- und Schnittstellenparadigma.....	113
4.3.5	Interoperabilität zwischen COM und .NET.....	116
5	Kritikpunkte und Anforderungen.....	119
5.1	Fallstudie: Projekt EPOS.....	119
5.2	Probleme der Integration von Optimierungssoftware in EUS.....	124
5.2.1	Allgemeine Probleme.....	124
5.2.2	Probleme spezifischer Optimierungssysteme.....	127
5.3	Anforderungen an Optimierungssoftware-Schnittstellen.....	129
5.3.1	Allgemeine Anforderungen an Software.....	130
5.3.2	Anforderungen an Programmierschnittstellen.....	131
5.3.3	Anforderungen an Endbenutzerschnittstellen.....	133
6	Konzeption einer Middleware für Optimierungssoftware.....	135
6.1	Kernel Layer.....	137
6.2	Model Management Layer.....	138
6.3	Prozedurale Schnittstellen.....	141
6.3.1	Kommandozeilenschnittstelle.....	141
6.3.2	Statische Libraries.....	142
6.3.2.1	IMR-Schnittstelle in MOPS 6.x.....	143
6.3.2.2	IMR-Schnittstellen ab MOPS 7.x.....	145
6.3.2.3	Konzept zur Weiterentwicklung.....	146
6.3.3	Dynamische Library.....	147
6.3.4	Kompatibilitätsbibliotheken.....	149
6.4	Objektorientierte Schnittstellen.....	153
6.4.1	Gemeinsame Modellierungsklassen.....	153
6.4.1.1	CMatrix, CModel und CSolver.....	154
6.4.1.2	Einfache Indexsets.....	155
6.4.1.3	Multi-Indexsets.....	156
6.4.1.4	Datenarrays.....	157
6.4.1.5	Variablen- und Restriktionsklassen.....	157
6.4.1.6	Lineare Ausdrücke.....	159
6.4.2	C++-Klassenbibliothek.....	160
6.4.2.1	Modellierung mit überladenen Operatoren.....	160
6.4.2.2	Ausnahmebehandlung.....	165
6.4.3	COM-Komponentenbibliothek.....	165
6.4.3.1	Überblick.....	166
6.4.3.2	Detailaspekte.....	168

6.4.3.3	Beschreibung der einzelnen COM-Komponenten.....	174
6.4.4	.NET-Komponentenbibliothek	183
6.5	OSI.....	185
6.6	Schnittstellen zu Modellierungssprachen	186
6.7	MOPS Studio als endbenutzerbezogene Schnittstelle.....	189
6.8	Optimierungsschnittstellen im Rahmen Service Orientierter Architekturen.....	192
7	Implementierung.....	197
7.1	Prozedurale Schnittstellen	197
7.1.1	DLL- und IMR-Interfaces	197
7.1.2	Internes Speicherpointermanagement.....	198
7.1.3	Callbacks	199
7.1.4	Interlanguage Interface	201
7.2	COM-Komponentenbibliothek.....	202
7.2.1	Verwendete Technologien.....	203
7.2.2	Interfaces der Hauptkomponente.....	203
7.2.3	Interne Modellspeicherung	205
7.2.4	Handling von VARIANTS	207
7.2.5	Interaktion zwischen Komponenten	208
7.3	Modellierungssprachenschnittstellen	209
7.3.1	AMPL	209
7.3.2	GNU MathProg	212
7.3.3	LINDO.....	212
7.3.4	MPL	212
7.4	MOPS Studio.....	215
8	Evaluation.....	219
8.1	Beispiele für Modellgeneratoren	219
8.1.1	Einfaches Produktionsmodell	219
8.1.2	Erweitertes Produktionsmodell	221
8.1.3	Diet-Modell	224
8.1.4	Excel-Unterstützung	226
8.2	Bewertung der Komponentenbibliothek.....	230
8.2.1	Bewertung nach allgemeinen Softwarequalitätsmerkmalen	230
8.2.2	Bewertung nach Erfüllung spezifischer Anforderungen	234
8.3	Bewertung von MOPS Studio	235
8.3.1	Bewertung nach allgemeinen Softwarequalitätsmerkmalen	235
8.3.2	Bewertung nach Erfüllung spezifischer Anforderungen	236
8.3.3	Bewertung des Einsatzes in der Lehre.....	237
9	Ausblick.....	239
	Literaturverzeichnis	241
	Anhang 1: Vergleich prozeduraler Solver-Funktionen	253
	Anhang 2: Klassenübersicht BCL 3.0	275
	Anhang 3: Klassenübersicht ILOG Concert Technology.....	278
	Anhang 4: Gemeinsame Modellierungsklassen	286
	Anhang 5: Interfaces der COM-Bibliothek	293
	Anhang 6: Prozedurales Model Manager Interface.....	300
	Lebenslauf	303
	Versicherung.....	305