

Inhaltsverzeichnis

1. Einleitung	1
1.1. Rahmenbedingungen	2
1.1.1. Aufgabenstellung	2
1.1.2. Voraussetzungen des Projekts	2
1.1.3. Planung und Ablauf des Projekts	5
1.1.4. Wissenschaftlicher und technischer Stand	6
1.1.5. Zusammenarbeit mit anderen Stellen	7
1.2. Übersicht des MINT-Ansatzes	8
1.3. Aufbau des Dokuments	9
2. Grundlagen	11
2.1. Modellgetriebene Software-Entwicklung	11
2.2. Integration von Softwaresystemen	13
2.2.1. Das Dublo-Muster	14
2.2.2. Der BALES-Ansatz	15
2.2.3. Serviceorientierte Architekturen	17
2.3. Modellierungswerzeuge	18
2.3.1. Eclipse Modeling Framework (EMF)	18
2.3.2. Graphical Modeling Framework (GMF)	18
2.4. Transformationswerkzeuge	19
2.4.1. Vorauswahl von MDA-Werkzeugen	19
2.4.2. AndroMDA	20
2.4.3. openArchitectureWare	21
2.4.4. Software Factories	24
2.4.5. Bewertung der untersuchten Werkzeuge	26
2.5. Werkzeuge für die modellgetriebene Datenintegration	28
2.5.1. SCORE Adaptive Bridges – Data Architecture Integration	28
2.5.2. HyperSenses	33
2.6. Visual Composer	36
2.6.1. Motivation	36
2.6.2. Einführung in SAP Visual Composer	37
2.6.3. Anwendungsszenario mit Visual Composer	40
2.6.4. Ergebnisse	43
2.7. Anwendungsszenarien für die Prozessintegration	45
2.7.1. Störungserfassungsdomäne	47
2.7.2. Instandhaltungsdomäne	47
2.8. Die andrena Testumgebung	49

Inhaltsverzeichnis

3. Domänen spezifische Modellierung mit der MINT-XL	53
3.1. Domänen spezifische Modellierung von Integrationsaspekten	53
3.2. MINT-XL	55
3.3. Erweiterungsmechanismen	58
3.3.1. Allgemeine Erweiterung: Mappings	59
3.3.2. Domänen spezifische Erweiterung: Störungsmanagement	60
3.4. Anwendungsbeispiel MINT-XL	62
3.5. Umsetzung	63
3.5.1. Umsetzung von MINT-XL mit EMF	63
3.5.2. Erstellung des MINT-PIE-Editors mit GMF	67
4. Umsetzung wissensintensiver Prozesse	71
4.1. Wissensintensive Prozesse	71
4.2. MINT Vorgehensmodell	72
4.3. Metamodellerzeugung	73
4.4. CIM	74
4.5. PIM	76
4.6. PSM	79
4.7. Zielsystem	80
4.8. Zusammenfassung	80
5. Modellgetriebene Prozessintegration	83
5.1. Modelle	84
5.1.1. Fachliches Modell	84
5.1.2. Plattformunabhängiges Architekturmodell	86
5.1.3. Annotationsmodell	87
5.1.4. Plattformabhängiges Architekturmodell	88
5.1.5. Altsystemmodell	89
5.1.6. Technisches Mappingmodell	92
5.1.7. Code	95
5.2. Transformationen	98
5.2.1. Vom fachlichen Modell zum plattformunabhängigen Architekturmodell	98
5.2.2. Vom fachlichen Modell zum technischen Mapping	100
5.2.3. Vom plattformunabhängigen Architekturmodell zum Annotationsmodell	102
5.2.4. Vom plattformunabhängigen zum plattformabhängigen Architekturmodell	103
5.2.5. Vom plattformabhängigen Architekturmodell zu Code	104
5.3. Vorgehen	106

6. Modellgetriebene Datenintegration	109
6.1. Persistenzadapter	109
6.2. Persistenzadapter-Generierung mit SCORE	110
6.3. Schema-Import	110
6.4. Die Datenzugriffsschicht	112
6.5. Die Service-Schicht	117
6.5.1. Mappings	121
6.6. Adapter-Generierung	127
6.6.1. Der generierte Zielcode	128
6.7. Zusammenfassung	130
7. Evaluierung der Ergebnisse	131
7.1. Das Vorgehen	131
7.2. Der Goal Question Metric (GQM)-Plan	134
7.2.1. GQM-Plan für Wartbarkeit	136
7.2.2. GQM-Plan für Performance	139
7.3. Ergebnisse der Evaluierung	141
7.3.1. Aufsetzen des Experiments	141
7.3.2. Ergebnisse für Goal 1: Wartbarkeit	141
7.3.3. Ergebnisse für Goal 2: Performance	145
7.4. Einschränkungen und Validität der Evaluierung	151
8. Entwurfsentscheidungsunterstützung beim Entwurf von Integrationssystemen	153
8.1. Best Practices und Anti-Patterns	153
8.1.1. Übersicht	153
8.1.2. Best Practices - allgemeine Erkenntnisse	153
8.1.3. Best Practices - Entwurfsmuster	154
8.1.4. Best practices für die verschiedenen Szenarien	156
8.1.5. Anti-Patterns für die verschiedenen Szenarien	158
8.2. Entwurfsentscheidungsmatrix	160
9. Fazit	165
A. Veröffentlichungen	169
Abbildungsverzeichnis	173
Tabellenverzeichnis	175
Literaturverzeichnis	183