



Inhaltsverzeichnis

Kurzfassung	i
Abstract	iii
Danksagung	v
1 Einleitung	1
1.1 Anwendungsszenarien	2
1.2 Anforderungen kollaborativer Prozesse	2
1.3 Beiträge der Arbeit	6
1.4 Grundlegende Begriffe	8
1.5 Aufbau der Arbeit	14
I Entwicklung einer kollaborativen Transaktionsschicht	17
2 Existierende Ansätze	19
2.1 Technologien zur Unterstützung kooperativer und kollaborativer Arbeitsprozesse	19
2.1.1 Versionskontrollsysteme	19
2.1.2 Workflow-Modelle	20
2.1.3 Groupware – kollaborative Editoren	20
2.1.4 Diskussion	21
2.1.5 Fazit der Diskussion	25
2.2 Erweiterte Transaktionsmodelle	25
2.2.1 Geschachtelte Transaktionen	25
2.2.2 Maßgeschneiderte Transaktionsmodelle	27
2.2.3 Dynamische Aktionen	30
	vii



2.2.4	Kombination mit Workflow-Modellen	32
2.2.5	Generische Transaktionsmodelle	33
2.2.6	Gegenüberstellung der Modelle und Anforderungen	33
2.3	Synchronisationskonzepte	35
2.4	Zusammenfassung	36
3	Entwurf einer kollaborativen Transaktionsschicht	39
3.1	Allgemeines Applikationsschichtenmodell	39
3.2	Die kollaborative Transaktionsschicht	41
3.3	Annahmen und Einschränkungen	42
3.4	Zusammenfassung	43
4	Formales Daten- und Operationsmodell	45
4.1	Semantisches Datenmodell	45
4.1.1	Graphenmodell	45
4.1.2	Darstellung als Objektmodell	47
4.2	Semantisches Operationsmodell	50
4.2.1	Leseoperationen	53
4.2.2	Schreiboperationen	55
4.2.3	Operationsfolgen	58
4.3	Zusammenfassung	60
5	Kollaboratives Transaktionsmodell	61
5.1	Grundlegende Architektur	61
5.2	Formale Spezifikation	64
5.2.1	Grundlagen	64
5.2.2	Geschachteltes Transaktionsmodell	67
5.2.3	Start- und Terminierungseigenschaften	69
5.2.4	Sichtbarkeits- und Isolationseigenschaften	71
5.2.5	Semantische Abhängigkeiten	74
5.2.6	Atomaritätseigenschaften semantischer Operationen	78
5.2.7	Atomaritätseigenschaften von Subtransaktionen der Ebene 2	79
5.2.8	Hierarchische und intertransaktionale Atomaritätseigenschaften	80
5.2.9	Dauerhaftigkeitseigenschaften	82



5.3	Automatische Transaktionsdefinition	83
5.3.1	Die AUTOCOMPLETE-Metrik	84
5.4	Qualitative Evaluierung der transaktionalen Konzepte	87
5.4.1	ACID-Eigenschaften	87
5.4.2	Automatische Transaktionsdefinition	94
5.4.3	Die AUTOCOMPLETE-Metrik	94
5.5	Zusammenfassung	97
II Entwurf einer transaktionalen Middleware		99
6 Ziele und Anforderungen		101
7 Verwandte Arbeiten		103
7.1	Parallele und verteilte Datenbanksysteme	103
7.2	Transaktionsmonitore	105
7.3	Transaktionen in Cloud-Umgebungen	105
7.4	Zusammenfassung	109
8 Architektur		111
8.1	Grundlegendes Komponentenmodell	111
8.2	Architektur der Client-Bibliothek	113
8.3	Schnittstelle zur Speicherschicht	114
8.3.1	Anforderungen	114
8.3.2	Analyse aktueller Speicherlösungen	115
8.3.3	Schnittstellenspezifikation	118
8.3.4	Die Schnittstelle zu Amazon S3	119
8.4	Architektur der Service-Bibliothek	120
8.4.1	Registrierung	122
8.4.2	Lokaler Transaktionsmanager	123
8.4.3	Globaler Transaktionsmanager	125
8.5	Verteilung und Betrieb der Middleware	126
8.6	Zusammenfassung	127



9	Implementierung des Transaktionsmodells	129
9.1	Clientseitige Integration	129
9.1.1	Annahmen bezüglich der Applikationslogik	130
9.1.2	Einem kollaborativen Szenario beitreten	133
9.1.3	Einhüllen logischer Leseoperationen in eine Subtransaktion der Ebene 2	134
9.1.4	Einhüllen logischer Änderungsoperationen in eine Subtransaktion der Ebene 2	136
9.1.5	Sichern durchgeführter Änderungen	139
9.1.6	Verwerfen durchgeführter Änderungen	141
9.2	Zentralisiertes Szenario	142
9.2.1	Starten der benötigten Transaktionen	142
9.2.2	Serviceseitige Umsetzung semantischer Operationen	144
9.2.3	Wahrung strikter Lesekonsistenz	146
9.2.4	Propagierung der durchgeführten Änderungen	147
9.2.5	Serviceseitiges Beenden von Transaktionen	149
9.2.6	Serviceseitiges Abrechnen von Transaktionen	150
9.3	Verteiltes Szenario	151
9.3.1	Anwendbarkeit des kollaborativen Transaktionsmodells in verteilten Umgebungen	151
9.3.2	Vereinfachungen des kollaborativen Transaktionsmodells	154
9.3.3	Teilnahme an verteilten kollaborativen Szenarien	156
9.3.4	Starten verteilter Transaktionen	159
9.3.5	Verteilte Ausführung semantischer Operationen	160
9.3.6	Verteilte Propagierung durchgeführter Änderungen und Beenden verteilter Transaktionen	160
9.3.7	Verwerfen durchgeführter Änderungen	162
9.4	Zusammenfassung	164
10	Mehrbenutzersynchronisation	167
10.1	Auswahl eines geeigneten Korrektheitskriteriums	167
10.2	Konfliktanalyse des semantischen Operationsmodells	168
10.3	Überblick über existierende Synchronisationsprotokolle	171
10.4	Umsetzung des 2-Phasen-Sperrprotokolls	173
10.4.1	Sperrmatrix	174



10.4.2 Sperrprotokoll	175
10.5 Umsetzung des optimistischen Synchronisationsprotokolls	181
10.5.1 Grundlegende Validierungsregel	181
10.5.2 Anpassung der Validierungsregel an das semantische Daten- und Operationsmodell	182
10.5.3 Abbildung des BOCC-Verfahrens auf das kollaborative Transaktionsmodell	185
10.6 Verteilte Synchronisation	186
10.6.1 Auswahl eines Korrektheitskriteriums	186
10.6.2 Gewährleistung globaler Serialisierbarkeit	189
10.6.3 Umsetzung pessimistischer und optimistischer Synchronisation in verteilten Szenarien	190
10.7 Zusammenfassung	193
11 Recovery	195
11.1 Fehlerklassen und Recovery-Klassen	195
11.2 Transaktions-Recovery	196
11.2.1 Verwaltung der Abbruchabhängigkeiten und Commit-Abhängigkeiten	197
11.2.2 Shadow Paging	198
11.2.3 Logging	199
11.2.4 Verteiltes Commit	203
11.2.5 Umgang mit den Fehlerklassen	204
11.3 System-Recovery	205
11.4 Zusammenfassung	206
12 Evaluierung	209
12.1 Testumgebung	210
12.1.1 Mikro-Benchmark	210
12.1.2 Systemumgebung	215
12.2 Kennzahlen	215
12.3 Parameter	216
12.4 Hypothesen	219
12.5 Nutzen und Kosten des Einsatzes semantischer Synchronisation	220
12.5.1 Messbasis	220



12.5.2	Abbruchrate unter Anwendung des BOCC-Protokolls	221
12.5.3	Neustartrate unter Anwendung pessimistischer Synchronisation	231
12.5.4	Transaktionsdurchsatz	238
12.6	Wahl der Fragmentgröße am Beispiel der Speicherlösung Amazon S3 . .	246
12.6.1	Schreib-/Lesegeschwindigkeit	246
12.6.2	Speicherkosten	248
12.6.3	Kosten strikter Lesekonsistenz	249
12.7	Zusammenfassung	250
13	Zusammenfassung	253
14	Ausblick	259
15	Anhang	263