

# Fortschritt-Berichte VDI

VDI

Reihe 10

Informatik/  
Kommunikation

Nr. 873

Dipl.-Ing. Stephan Hensel  
Dresden

## Semantische Revisions- kontrolle für die Evolution von Informations- und Datenmodellen

Berichte aus der Professur für **Prozessleittechnik**  
und der Arbeitsgruppe **Systemverfahrenstechnik**  
der TU Dresden, Prof. Dr.-Ing. habil. Leon Urbas (Hrsg.)



TECHNISCHE  
UNIVERSITÄT  
DRESDEN



## **Semantische Revisionskontrolle für die Evolution von Informations- und Datenmodellen**

Semantic revision control for the evolution of information and  
data models

Dipl.-Ing. Stephan Hensel

Der Fakultät Elektrotechnik und Informationstechnik  
der Technischen Universität Dresden

zur Erlangung des akademischen Grades eines

Doktoranden

(Dr.-Ing.)

genehmigte  
**Dissertation**

Vorsitzender: Prof. Dr.-Ing. Dr. h.c. Frank H. P. Fitzek

Gutachter: Prof. Dr.-Ing. habil. Leon Urbas

Gutachter: Univ.-Prof. Dr.-Ing. Alexander Fay

Gutachter: Prof. Dr.-Ing. Christian Diedrich

Tag der Einreichung: 16.05.2019

Tag der Verteidigung: 20.11.2020



# Fortschritt-Berichte VDI

Reihe 10

Informatik/  
Kommunikation

Dipl.-Ing. Stephan Hensel,  
Dresden

Nr. 873

Semantische Revisions-  
kontrolle für die  
Evolution von  
Informations- und  
Datenmodellen

Berichte aus der Professur für **Prozessleittechnik**  
und der Arbeitsgruppe **Systemverfahrenstechnik**  
der TU Dresden, Prof. Dr.-Ing. habil. Leon Urbas (Hrsg.)



Hensel, Stephan

## **Semantische Revisionskontrolle für die Evolution von Informations- und Datenmodellen**

Fortschr.-Ber. VDI Reihe 10 Nr. 873. Düsseldorf: VDI Verlag 2021.

188 Seiten, 64 Bilder, 5 Tabellen.

ISBN 978-3-18-387310-4, ISSN 0178-9627,

€ 67,00/VDI-Mitgliederpreis € 60,30.

**Keywords:** Semantik – Revisionskontrolle – Evolution – Informationsmodelle – Datenmodelle – Linked Data – R43ples – Co-Simulation – Modularisierung – Module Type Package

Im Rahmen dieser Dissertation wurde ein Revision Management System zur durchgängigen Unterstützung der Evolution von Informations- und Datenmodellen entwickelt, das Revisionsverwaltungs- und Evolutionsmechanismen integriert. Besonderheit ist hierbei die technologieunabhängige mathematische und semantische Beschreibung, die eine Überführung des Konzepts in unterschiedliche Technologien ermöglicht. Beispielhaft wurde das Konzept für das Semantic Web als Weiterentwicklung des Open-Source-Projektes R43ples umgesetzt.

### **Bibliographische Information der Deutschen Bibliothek**

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliographische Daten sind im Internet unter [www.dnb.de](http://www.dnb.de) abrufbar.

### **Bibliographic information published by the Deutsche Bibliothek**

(German National Library)

The Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliographie (German National Bibliography); detailed bibliographic data is available via Internet at [www.dnb.de](http://www.dnb.de).

© VDI Verlag GmbH · Düsseldorf 2021

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe (Fotokopie, Mikrokopie), der Speicherung in Datenverarbeitungsanlagen, im Internet und das der Übersetzung, vorbehalten.

Als Manuskript gedruckt. Printed in Germany.

ISSN 0178-9627

ISBN 978-3-18-387310-4

## Danksagung

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich in der Zeit als wissenschaftlicher Mitarbeiter und Doktorand an der Professur für Prozessleittechnik und Arbeitsgruppe Systemverfahrenstechnik fachlich und persönlich unterstützt haben.

Ein besonderer Dank gilt Herrn Prof. Dr.-Ing. habil. Leon Urbas, meinem Doktorvater, für die Betreuung dieser Arbeit und die Bereitstellung der notwendigen Infrastruktur für die Umsetzung der Arbeit in einem sehr spannenden Themenkomplex. Durch die zahlreichen Projekte hat er mir viele Einblicke in die industrielle Praxis ermöglicht, wodurch ich auch wichtige Kontakte für meine weitere berufliche Zukunft knüpfen konnte. Seine kritischen Nachfragen und die gemeinsamen wissenschaftlichen Diskussionen haben wesentlich zur Ideenfindung für diese Dissertation beigetragen.

Ich danke Herrn Univ.-Prof. Dr.-Ing. Alexander Fay für die sehr gute Zusammenarbeit und die Übernahme der Zweitbegutachtung.

Meiner Familie möchte ich dafür danken, dass sie mich während all der Zeit so herzlich unterstützt hat und immer verständnisvoll war, dass die Erstellung der Dissertation viel Zeit in Anspruch genommen hat.

Weiterhin danke ich den zahlreichen Korrekturleserinnen und Korrekturlesern, die sehr viel Zeit in die Überprüfung von Rechtschreibung, Kommasetzung und vielen sprachlichen Kleinigkeiten investiert haben.

Außerdem möchte ich mich bei meinen Kollegen für die gute Zeit am Lehrstuhl, die vielen gemeinsamen Dienstreisen und den intensiven wissenschaftlichen Diskurs bedanken.

Schließlich danke ich Dr. Jürgen Hambrecht und Eggert Voscherau, den Gründern der HaVo-Stiftung, für die finanzielle Unterstützung im Rahmen eines HaVo-Stipendiums.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Zielstellung und erwartete Ergebnisse . . . . .	2
1.2.1	Kernthese . . . . .	2
1.2.2	Einzelthesen . . . . .	2
1.3	Einordnung und Abgrenzung der Arbeit . . . . .	3
1.4	Anwendungsfälle . . . . .	4
1.4.1	Co-Simulation . . . . .	4
1.4.2	Modularisierung . . . . .	5
1.5	Gliederung der Arbeit . . . . .	6
<b>2</b>	<b>Grundlagen</b>	<b>8</b>
2.1	Aspekte der Veränderlichkeit . . . . .	8
2.2	Informationsmodellierung . . . . .	10
2.2.1	Terminologie . . . . .	10
2.2.1.1	Informationsmodell und Semantik . . . . .	10
2.2.1.2	Informationsraum . . . . .	12
2.2.1.3	Arten von Informationsmodellen . . . . .	13
2.2.1.4	Ontologie . . . . .	14
2.2.2	Lebenszyklus . . . . .	14
2.2.3	Vernetzung innerhalb eines Informationsraums . . . . .	15
2.3	Evolution . . . . .	15
2.3.1	Terminologie . . . . .	15
2.3.1.1	Evolution . . . . .	16
2.3.1.2	Co-Evolution . . . . .	16
2.3.1.3	Evolvability . . . . .	18
2.3.1.4	Wartung und Wartbarkeit . . . . .	19
2.3.2	Evolution in verwandten Themengebieten . . . . .	19
2.3.2.1	Schema-Evolution . . . . .	20
2.3.2.2	Ontologie-Evolution . . . . .	20
2.3.2.3	Schema-Evolution vs. Ontologie-Evolution . . . . .	21
2.4	Revisionsverwaltung . . . . .	22
2.4.1	Terminologie . . . . .	23
2.4.1.1	Revisionsverwaltung vs. Versionsverwaltung . . . . .	23
2.4.1.2	Basisbegriffe der Revisionsverwaltung . . . . .	24
2.4.1.3	Arten von Revisionsverwaltung . . . . .	24
2.4.1.4	Synchronisation und Replikation . . . . .	25
2.4.1.5	Verfahren zur Konsistenzerhaltung . . . . .	26

2.4.2	Erweiterte Revisionskontrolle für Modelldaten . . . . .	26
2.5	Konsistenz . . . . .	27
2.5.1	Terminologie . . . . .	28
2.5.1.1	Konsistenz . . . . .	28
2.5.1.2	Klassifikation von Modellkonsistenz . . . . .	30
2.5.2	CAP-Theorem . . . . .	31
<b>3</b>	<b>Analyse</b>	<b>32</b>
3.1	Anforderungsanalyse . . . . .	32
3.1.1	Prinzipien mit Einfluss auf Evolvability . . . . .	32
3.1.1.1	P1 - Entwicklung von stabilen Zwischenergebnissen (X,-)	33
3.1.1.2	P2 - Nutzung von evolutionärer Entwicklung (X,\$)	33
3.1.1.3	P3 - Verständnis des Unternehmens (X,\$)	33
3.1.1.4	P4 - Bereitstellung von überprüfbarer Zuständen (x,\$)	34
3.1.1.5	P5 - Nutzung von offenen Standards (x,-)	34
3.1.1.6	P6 - Identifizierung von Dingen, die sich wahrscheinlich ändern (X,-)	34
3.1.1.7	P7 - Design für Evolvability (X,\$)	35
3.1.2	Technologische Sicht . . . . .	35
3.1.2.1	Nutzungskontext . . . . .	35
3.1.2.2	Änderungsmanagement . . . . .	36
3.1.2.3	Evolution . . . . .	38
3.1.2.4	Semantische Modellbeschreibung . . . . .	39
3.1.2.5	Qualitätsattribute . . . . .	39
3.1.3	Anwendungsfälle . . . . .	40
3.1.3.1	Co-Simulation . . . . .	40
3.1.3.2	Modularisierung . . . . .	44
3.1.4	Anforderungen . . . . .	47
3.2	Analyse bestehender Ansätze . . . . .	51
3.2.1	Dissertation Timo Kehrer [Keh15] . . . . .	51
3.2.2	Dissertation Ljiljana Stojanovic [Sto04] . . . . .	51
3.2.3	SecVolution . . . . .	52
3.2.4	Simantics . . . . .	53
3.2.5	Changes Tab . . . . .	53
3.2.6	R43ples . . . . .	55
3.2.7	Zusammenfassung . . . . .	56
3.3	Analyseergebnisse und Priorisierung . . . . .	58
<b>4</b>	<b>Entwurf</b>	<b>59</b>
4.1	Lebenszyklusmodell für Informationsmodelle . . . . .	59
4.2	Revision Management System . . . . .	61
4.2.1	Komponentenübersicht . . . . .	61
4.2.2	Data Management . . . . .	63
4.2.3	Control . . . . .	65

4.2.4	User Interface . . . . .	67
4.3	Formale Beschreibung verbindungsorientierter Modelle . . . . .	67
4.3.1	Compound Graphs . . . . .	68
4.3.2	Compound Graphs Erweiterung . . . . .	70
4.3.3	Semantische Beschreibung . . . . .	71
4.4	Änderungsmanagement . . . . .	72
4.4.1	Revisionskontrolle . . . . .	73
4.4.1.1	Revisionsgraph . . . . .	73
4.4.1.2	Vorgänger-/Nachfolgerbeziehungen . . . . .	74
4.4.1.3	Pfadgenerierung und Deltawiederherstellung . . . . .	75
4.4.1.4	Grundlegende Revisionskontrollfunktionalitäten . . . . .	76
4.4.1.5	Semantische Beschreibung . . . . .	80
4.4.2	Aggregation von High-Level-Changes . . . . .	81
4.4.2.1	Mathematische Beschreibung . . . . .	82
4.4.2.2	Semantische Beschreibung . . . . .	82
4.4.3	Zusammenführung divergierter Entwicklungszweige . . . . .	83
4.4.3.1	Methoden der Zusammenführung . . . . .	83
4.4.3.2	Konflikterkennung und -behebung . . . . .	86
4.4.3.3	Semantische Beschreibung . . . . .	93
4.5	Evolutions- und Konsistenzmechanismen . . . . .	96
4.5.1	Evolutionsmechanismen . . . . .	96
4.5.1.1	Integration in RMS . . . . .	96
4.5.1.2	Mathematische Beschreibung . . . . .	98
4.5.1.3	Semantische Beschreibung . . . . .	100
4.5.2	Konsistenzmechanismen . . . . .	101
<b>5</b>	<b>Implementierung</b>	<b>103</b>
5.1	Übersicht . . . . .	103
5.2	Änderungsmanagement . . . . .	106
5.2.1	Ontologie . . . . .	106
5.2.2	Basisrevisionskontrollfunktionalitäten . . . . .	108
5.2.3	Aggregation von High-Level-Changes . . . . .	110
5.2.4	Zusammenführung divergierter Entwicklungszweige . . . . .	113
5.3	Evolutionsmechanismen . . . . .	116
5.4	Weitere Arbeiten in diesem Bereich . . . . .	118
<b>6</b>	<b>Verifikation</b>	<b>119</b>
6.1	Beispielhafte Nutzung der formalen Beschreibung . . . . .	119
6.2	Nachweis der Erzeugung von beliebigen Revisionsinhalten . . . . .	125
6.3	Abbildung verbindungsorientierter Modelle am Beispiel der Co-Simulation	127
6.4	Testfälle innerhalb der Implementierung . . . . .	130
<b>7</b>	<b>Diskussion</b>	<b>132</b>
7.1	Methodikbewertung . . . . .	132

7.2	Ergebnisdiskussion und Verifikation der Thesen . . . . .	133
<b>8</b>	<b>Zusammenfassung</b>	<b>136</b>
8.1	Ergebniszusammenfassung . . . . .	136
8.2	Ausblick und Grenzen . . . . .	136
<b>Anhang A</b>	<b>Entwurf</b>	<b>141</b>
<b>Anhang B</b>	<b>Implementierung</b>	<b>142</b>
B.1	Basisrevisionskontrollfunktionalitäten . . . . .	142
B.2	Aggregation von High-Level-Changes . . . . .	146
B.3	Zusammenführung von divergierten Entwicklungszweigen . . . . .	148
B.4	Co-Evolution . . . . .	150
<b>Literaturverzeichnis</b>		<b>152</b>

# Abkürzungs- und Symbolverzeichnis

## Abkürzungen

<b>ACID</b>	Atomicity, Consistency, Isolation, Durability
<b>AERO</b>	Aggregation and Evolution Rules Ontology
<b>AutomationML</b>	Automation Markup Language
<b>BASE</b>	Basically Available, Soft state, Eventual consistency
<b>BPMN</b>	Business Process Model and Notation
<b>CAP</b>	Consistency, Availability, Partition Tolerance
<b>CHAO</b>	Change and Annotation Ontology
<b>CIF</b>	Continuous Integration Framework
<b>CPS</b>	Cyber Physical Systems
<b>CVS</b>	Concurrent Versions System
<b>DIMA</b>	Dezentrale Intelligenz für modulare Anlagen
<b>EMF</b>	Eclipse Modeling Framework
<b>FMI</b>	Functional Mock-up Interface
<b>FMU</b>	Functional Mock-up Unit
<b>GLD</b>	Goverment Linked Data
<b>IP</b>	Internetprotokoll
<b>KAON</b>	Karlsruhe Ontology and Semantic Web framework
<b>LDAP</b>	Lightweight Directory Access Protocol
<b>LED</b>	Linked Enterprise Data
<b>LOD</b>	Linked Open Data
<b>MDE</b>	Model-driven Engineering
<b>MMO</b>	Merge Management Ontology

<b>MOF</b>	Meta Object Facility
<b>MTP</b>	Module Type Package
<b>OPC UA</b>	Open Platform Communications Unified Architecture
<b>OWL</b>	Web Ontology Language
<b>PEA</b>	Process Equipment Assembly
<b>PFE</b>	Prozessführungsebene
<b>POL</b>	Process Orchestration Layer
<b>PROV-O</b>	PROV Ontology
<b>QUDT</b>	Quantities, Units, Dimensions, and Data Types
<b>RCS</b>	Revision Control System
<b>RDF</b>	Resource Description Framework
<b>RDFS</b>	Resource Description Framework Schema
<b>RMO</b>	Revision Management Ontology
<b>RMS</b>	Revision Management System
<b>SCCS</b>	Source Code Control System
<b>SHACL</b>	Shapes Constraint Language
<b>SPARQL</b>	SPARQL Protocol And RDF Query Language
<b>SPIN</b>	SPARQL Inferencing Notation
<b>SVN</b>	Apache Subversion
<b>TGG</b>	Tripel-Graph-Grammatik
<b>UML</b>	Unified Modeling Language
<b>URI</b>	Uniform Resource Identifier
<b>USB</b>	Universal Serial Bus
<b>VIBN</b>	Virtuelle Inbetriebnahme
<b>XML</b>	Extensible Markup Language

## Symbole und Funktionen

### Allgemein

$\mathbb{N}^+$	Menge der natürlichen Zahlen größer Null
$\mathbb{N}_0^+$	Menge der natürlichen Zahlen größer gleich Null
$ \mathcal{X} $	Mächtigkeit der Menge $\mathcal{X}$
$\mathcal{P}(M)$	Potenzmenge einer Menge $M$

### Compound Graphs

$\tilde{G}$	Compound Graph
$\tilde{G}'$	Einfach gerichteter Graph innerhalb des Compound Graphs
$\tilde{T}'$	Baum innerhalb des Compound Graphs
$\hat{G}$	Einfach gerichteter Graph
$\hat{T}$	Baum
$\hat{V}$	Menge an Knoten
$\hat{E}$	Menge an Kanten
$\hat{v}, \hat{w}$	Knoten aus $\hat{V}$
$\text{pred}_{\hat{G}}(\hat{v})$	Funktion zur Ermittlung von Vorgängern in $\hat{G}$
$\text{succ}_{\hat{G}}(\hat{v})$	Funktion zur Ermittlung von Nachfolgern in $\hat{G}$
$\tilde{B}$	Menge der Basisknoten (Blättern von $\tilde{T}'$ )
$\tilde{S}$	Menge der Subgraphen (innere Knoten von $\tilde{T}'$ )
$\tilde{V}$	Menge der Knoten als Vereinigung von $\tilde{B}$ und $\tilde{S}$
$\tilde{b}, \tilde{s}, \tilde{v}, \tilde{w}$	Knoten aus $\tilde{V}$
$\tilde{E}_{\tilde{G}'}$	Adjazenzkanten
$\tilde{E}_{\tilde{T}'}$	Inklusionskanten
$\tilde{n}_v$	Eindeutiger Identifikator des Knotens
$\tilde{n}_z$	Eindeutiger Identifikator zur Zuordnung der Semantik zum Knoten

## Änderungsmanagement

$\mathcal{S}$	Menge aller Statements
$\mathcal{G}$	Revisionsgraph
$\mathcal{R}$	Menge aller möglichen Revisionen
$\mathcal{C}$	Menge aller möglichen Änderungen zwischen zwei Revisionen (ChangeSets)
$\mathcal{B}$	Menge aller möglichen Entwicklungszweige (Branches)
$\mathcal{T}$	Menge aller möglichen Tags
$R_g$	Menge der Revisionen innerhalb von $\mathcal{G}$
$C_g$	Menge der Änderungen zwischen zwei Revisionen (ChangeSets) innerhalb von $\mathcal{G}$
$B_g$	Menge der Entwicklungszweige (Branches) innerhalb von $\mathcal{G}$
$T_g$	Menge der Tags innerhalb von $\mathcal{G}$
$n_g$	Eindeutiger Identifikator eines Revisionsgraphen in der Menge der Revisionsgraphen
$C^+$	Menge der hinzugefügten Elemente
$C^-$	Menge der gelöschten Elemente
$r_l$	Blattrevision des Entwicklungszweiges
$r_x, r_y, r_t$	Revisionen aus $\mathcal{R}$
$R_b$	Menge der Revisionen eines Entwicklungszweiges
$n_b$	Eindeutiger Identifikator eines Branches im Revisionsgraphen
$n_t$	Eindeutiger Identifikator eines Tags im Revisionsgraphen
$\Upsilon_l$	Vollständiger Revisionsinhalt des Blattes des Entwicklungszweiges
$\Upsilon_t$	Vollständiger Revisionsinhalt eines Tags
$\text{pred}_{\mathcal{G}}(r_y)$	Funktion zur Ermittlung von Vorgängerrevisionen in $\mathcal{G}$
$\text{succ}_{\mathcal{G}}(r_x)$	Funktion zur Ermittlung von Nachfolgerrevisionen in $\mathcal{G}$
$\text{path}_{\mathcal{G}}(r_x, r_y)$	Funktion zur Ermittlung eines Revisionspfades in $\mathcal{G}$
$\text{getContent}_{\mathcal{G}}(r_x)$	Funktion zur Wiederherstellung des vollständigen Revisionsinhalts

***strip*** ( $\Upsilon_x, C^+, C^-$ ) Funktion zur Berechnung von minimalen ChangeSets

$\Gamma$	Menge der Revisionsgraphen
<b><i>create</i></b> $_{\Gamma}$	Funktion zur Erstellung eines neuen Revisionsgraphen
<b><i>drop</i></b> $_{\Gamma}$	Funktion zur Löschung eines bestehenden Revisionsgraphen
<b><i>branch</i></b> $_{\mathcal{G}}$	Funktion zur Erstellung eines neuen Entwicklungszweiges
<b><i>tag</i></b> $_{\mathcal{G}}$	Funktion zur Erstellung eines neuen Tags
$\Upsilon_x$	Vollständiger Revisionsinhalt einer spezifizierten Revision
$r^*$	Neu erstellte Revision
<b><i>commit</i></b> $_{\mathcal{G}}$	Funktion zur Erstellung eines neuen Commits
<b><i>revert</i></b> $_{\mathcal{G}}$	Funktion, um einen vorher erstellten Commit rückgängig zu machen

### Aggregation von High-Level-Changes

<b><i>hlcAgg</i></b> $_{\mathcal{G}}$	Funktion zur Aggregation von atomaren Änderungen zu High-Level-Changes
$\Phi_{\mathcal{G}}$	Funktion zur Berechnung von High-Level-Changes zwischen zwei Revisionen
$n_z$	Eindeutiger Identifikator des High-Level-Changes
$C_r^+$	Menge der nicht zuzuordnenden hinzugefügten Elemente
$C_r^-$	Menge der nicht zuzuordnenden gelöschten Elemente

### Zusammenführung divergierter Entwicklungszweige

<b><i>merge</i></b> $_{\mathcal{G}}$	Funktion zur Zusammenführung von divergierten Entwicklungszweigen mittels eines 3-Wege-Merges
$b_s$	Quellentwicklungszweig
$b_t$	Zielentwicklungszweig
$\Psi_G$	Funktion zur Berechnung der ChangeSets in Bezug auf die jeweiligen Entwicklungszweige
<b><i>pick</i></b> $_{\mathcal{G}}$	Funktion zur Wiederverwendung von bereits durchgeföhrten Änderungen in Bezug auf eine Revision
$r_p$	Revision, die wiederverwendet werden soll

$\vec{R}_p$	Geordnete Liste von Revisionen
$pick_{\mathcal{G}}\left(\vec{R}_p, n_b\right)$	Funktion zur Wiederverwendung einer Liste an bestehenden Revisionen
$fastForward_{\mathcal{G}}$	Funktion zur Zusammenführung von divergierten Entwicklungszweigen, wobei Revisionshistorie geglättet wird und nicht notwendige 3-Wege-Merges vermieden werden
$\mathcal{K}$	Menge der möglichen Status von atomaren Änderungen
$getCommonAncestor_{\mathcal{G}}\left(n_{b_s}, n_{b_t}\right)$	Funktion zur Berechnung der gemeinsamen Vorgängerrevision von zwei Entwicklungszweigen
$r_c$	Gemeinsame Vorgängerrevision von zwei Entwicklungszweigen
$C_{path_s}$	Pfad von gemeinsamer Vorgängerrevision zum Blatt des Quellentwicklungszweiges
$C_{path_t}$	Pfad von gemeinsamer Vorgängerrevision zum Blatt des Zielentwicklungszweiges
$\Omega_{Start}$	Startmenge für das Nachvollziehen der Änderungen
$add(\Omega, s)$	Funktion zur Aktualisierung des Status eines Statements, wenn dieses hinzugefügt wird
$del(\Omega, s)$	Funktion zur Aktualisierung des Status eines Statements, wenn dieses gelöscht wird
$s$	Einzelnes Statement
$F_{C_n}(\Omega)$	Funktion zu Anwendung der Aktualisierungen von einer Revision
$\Omega_{End}$	Endmenge mit allen Status nach dem Nachvollziehen der Änderungen
$\mathcal{Q}$	Menge an Definitionen zur automatisierten Konfliktbehebung
$k_s$	Status Quellentwicklungszweig
$k_t$	Status Zielentwicklungszweig
$q$	Boolesche Größe zur Spezifikation, ob es sich um einen Konflikt handelt oder nicht
$\mathcal{D}_s$	Menge der Status auf dem Quellentwicklungszweig ohne gleiche Status des Zielentwicklungszweiges
$\mathcal{D}_t$	Menge der Status auf dem Zielentwicklungszweig ohne gleiche Status des Quellentwicklungszweiges

$\tilde{\mathcal{D}}_s$	Menge der Status auf dem Quellentwicklungs zweig mit gleicher Kardinalität, wie Zielentwicklungs zweig
$\tilde{\mathcal{D}}_t$	Menge der Status auf dem Zielentwicklungs zweig mit gleicher Kardinalität, wie Quellentwicklungs zweig
$\mathcal{D}_{Diff}$	Zusammenführung der Mengen $\tilde{\mathcal{D}}_s$ und $\tilde{\mathcal{D}}_t$ zur Beschreibung der Unterschiede
$\mathcal{D}_Q$	Ergebnismenge der Konflikterkennung und -behebung mit entsprechenden Status für eine automatisierte Konfliktbehebung
<b><math>hlcPathAgg</math></b> ( $C_{path}$ )	Funktion zur Aggregation von atomaren Änderungen zu High-Level-Changes entlang eines Pfades
<b><math>intersecg</math></b> ( $C_{path}$ )	Funktion zur Berechnung von Abhängigkeiten von High-Level-Changes

## Evolutionsmechanismen

$\mathcal{G}_e$	Revisionsgraph zur semantischen Beschreibung von durchgeföhrten Co-Evolutionen
$h_{\mathcal{G}_s}$	Ergebnis der High-Level-Change-Aggregation der zu co-evolvierenden Änderungen
<b><math>calcDepr</math></b> ( $\mathcal{G}_s, r_{x_s}, r_{y_s}, h_{\mathcal{G}_s}$ )	Funktion zur Berechnung von Abhängigkeiten
$\mathcal{G}_s$	Quellrevisionsgraph
$\mathcal{G}_t$	Zielrevisionsgraph
<b><math>coevolver</math></b> ( $h_{\mathcal{G}_s}, \mathcal{G}_t, n_{b_t}$ )	Funktion zur Erstellung von Co-Evolutionscommits auf abhängige Entwicklungszweige
$\mathcal{E}(h_{\mathcal{G}_s}, \mathcal{G}_t, n_{b_t})$	Funktion zur Berechnung der Hinzufügungen und Löschen für die Co-Evolution
<b><math>coevolveAll_{\Gamma}</math></b> ( $\mathcal{G}_s, r_{x_s}, r_{y_s}$ )	Funktion zur Co-Evolution von allen abhängigen Revisionsgraphen und Entwicklungszweigen
$\mathcal{Z}$	Menge zur temporären Speicherung der durchgeföhrten Änderungen
$\Xi(\mathcal{Z})$	Funktion zur Überführung der durchgeföhrten Änderungen in Hinzufügungen und Löschungen zur semantischen Beschreibung

## Implementierung

$\mathcal{L}$	Menge aller Literale
$\mathcal{U}$	Menge aller Uniform Resource Identifiers

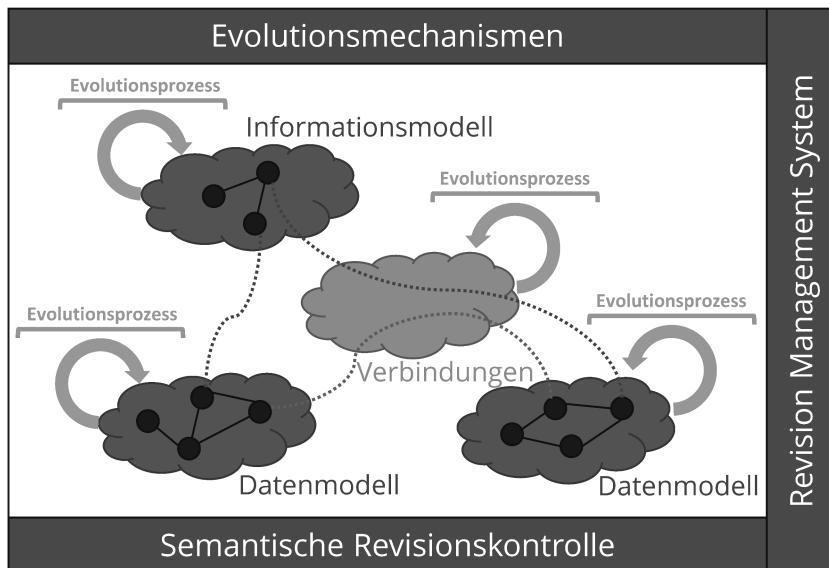
## Verifikation

$\tilde{I}$	Menge der Eingangssports
$\tilde{O}$	Menge der Ausgangssports
$d_{\tilde{G}'}^-(\tilde{v})$	Eingangsgrad eines Knotens $\tilde{v}$ in $\tilde{G}'$
$d_{\tilde{G}'}^+(\tilde{v})$	Ausgangsgrad eines Knotens $\tilde{v}$ in $\tilde{G}'$
$\tilde{E}_{\tilde{G}'_{connection}}$	Menge der Verbindungen über Komponenten hinweg
$\tilde{E}_{\tilde{G}'_{dependency}}$	Menge der internen Abhängigkeiten innerhalb einer Komponente

## Kurzfassung

### Semantische Revisionskontrolle für die Evolution von Informations- und Datenmodellen

Stärker verteilte Systeme in der Planung und Produktion verbessern die Agilität und Wartbarkeit von Einzelkomponenten, wobei gleichzeitig jedoch deren Vernetzung untereinander steigt. Das stellt wiederum neue Anforderungen an die semantische Beschreibung der Komponenten und deren Verbindungen, wofür Informations- und Datenmodelle unabdingbar sind. Der Lebenszyklus dieser Modelle ist dabei von Änderungen geprägt, mit denen umgegangen werden muss. Heutige Revisionsverwaltungssysteme, die die industriell geforderte Nachvollziehbarkeit bereitstellen könnten, sind allerdings nicht auf die speziellen Anforderungen der Informations- und Datenmodelle zugeschnitten, wodurch Möglichkeiten einer konsistenten Evolution verringert werden.

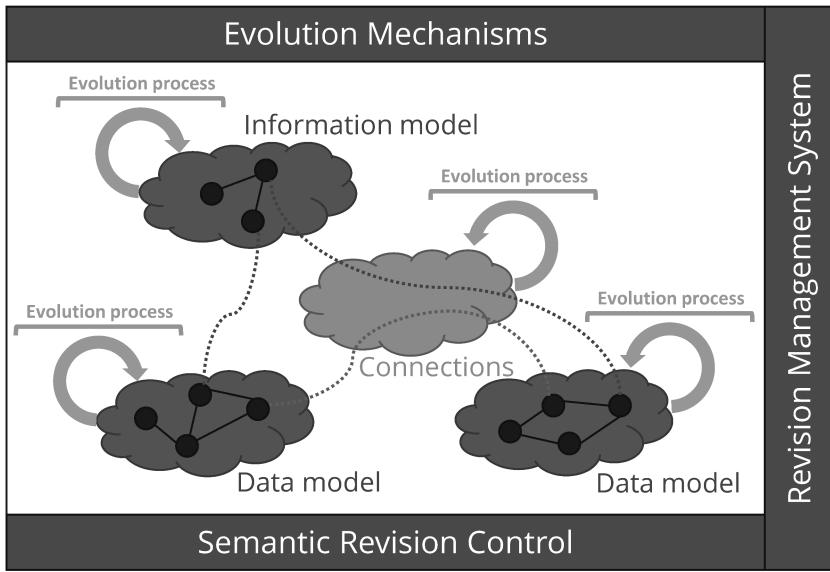


Im Rahmen dieser Dissertation wurde ein Revision Management System zur durchgängigen Unterstützung der Evolution von Informations- und Datenmodellen entwickelt, das Revisionsverwaltungs- und Evolutionsmechanismen integriert. Besonderheit ist hierbei die technologieunabhängige mathematische und semantische Beschreibung, die eine Überführung des Konzepts in unterschiedliche Technologien ermöglicht. Beispielhaft wurde das Konzept für das Semantic Web als Weiterentwicklung des Open-Source-Projektes R43ples umgesetzt.

## Abstract

### Semantic revision control for the evolution of information and data models

The increased distribution of systems in planning and production leads to improved agility and maintainability of individual components, whereas concurrently their cross-linking increases. This causes new requirements for the semantic description of components and links for which information and data models are indispensable. The life cycle of those models is characterized by changes that must be dealt with. However, today's revision control systems would provide the required industrial traceability but are not enough for the specific requirements of information and data models. As a result, possibilities for a consistent evolution are reduced.



Within this thesis a revision management system was developed, integrating revision control and evolution mechanisms to support the evolution of information and data models. The key is the technology-independent mathematical and semantic description allowing the application of the concept within different technologies. Exemplarily the concept was implemented for the Semantic Web as an extension of the open source project R43ples.