Diplomarbeit Nr. 2203

Generische Modellierung und Aktualisierung von Vertrauen im UniTEC Reputationssystem

Ernesto Baschny

Studiengang:	Informatik
Prüfer:	Prof. Dr. rer. nat. Dr. h. c. Kurt Rothermel
Betreuer:	Dipl.-Inf. Michael Kinateder
begonnen am:	19. April 2004
beendet am:	18. Oktober 2004
CR-Klassifikation:	J.4, I.2.6, E.2, I.6.5, K.4.4

Institut für Parallele und
Verteilte Systeme (IPVS)
Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

UniTEC ist ein verteiltes Reputationssystem, in dem unterschiedliche Benutzer in einem Peer-to-Peer Netz miteinander kommunizieren können. Der Grad und die Art der Kooperation zwischen beteiligten Benutzern hängen von dem Vertrauensverhältnis an, das zwischen ihnen aufgebaut wurde. Um dieses in einem Rechnersystem umsetzen zu können, muss zunächst das menschliche Vertrauensverständnis in ein mathematisches Modell übertragen werden. Die Vertrauensverhältnisse sollen nicht explizit durch den Benutzer eingegeben werden, sondern sollen aufgrund von Beobachtungen, Erfahrungen und Aussagen Dritter implizit berechnet werden. Dazu dienen Vertrauensalgorithmen, die diese unterschiedlichen Eingaben innerhalb des Vertrauensmodells umsetzten können.

Nach einer Einführung zu dem Thema Vertrauen folgen die Ergebnisse dieser Diplomarbeit. Die Modellierung von Vertrauen steht dabei im Mittelpunkt. Es wird eine Auswahl von Vertrauensmodellen und Vertrauensalgorithmen aus der Literatur untersucht, und schließlich ein generisches Vertrauensmodell erarbeitet, mit dem das UniTEC-System arbeiten kann. Das Generische an diesem Modell erlaubt, dass andere Vertrauensmodelle darauf aufbauend ebenfalls im UniTEC-System integriert werden können. Die ausgewählten Vertrauensmodelle und Algorithmen aus der Literatur werden umgesetzt, wenn notwendig erweitert, getestet und schließlich einer umfangreichen Analyse unterzogen.

Inhaltsverzeichnis

1. Einführung **13**
1.1. Motivation 13
1.2. Struktur der Arbeit 14

2. Das Konzept „Vertrauen“ **15**
2.1. Vertrauensbegriff 15
2.1.1. Vertrauensabsicht 17
2.1.2. Reputation 17
2.2. Kooperation 18
2.3. Vertrauenszustände 18
2.4. Vertrauensdynamik 19
2.5. Kontexte 20
2.6. Verhaltensmuster 21

3. Bestandteile von Vertrauensmodellen und -algorithmen **23**
3.1. Vertrauensmetrik 23
3.2. Vertrauensgewissheit 24
3.3. Vertrauensaktualisierungsereignisse 24
3.4. Gewichtung einer Vertrauensaktualisierung 25
3.5. Kontexte 25
3.6. Vertrauensziel 26
3.7. Erfahrungen (Bewertungen von Interaktionen) 26
3.8. Alterung 27
3.8.1. Alterungsfaktor 28
3.8.2. Bewertungshistorienfenster 28
3.9. Abnutzung 28
3.10. Vertrauensaktualisierungsalgorithmen 29
3.11. Erwartungen 29
3.11.1. Umfang 29
3.11.2. Vertrauensinitialisierung 29
3.11.3. Verhalten der Vertrauensmetrik 30
3.11.4. Verhalten der Vertrauensgewissheit 30

4. Vertrauensmodelle und -algorithmen aus der Literatur 31
4.1. Kriterien zur Auswahl . . . 31
4.2. Vertrauensmodelle . . . 31
4.2.1. Jonker [1999] . . . 32
4.2.2. Bin Yu [2000] . . . 33
4.2.3. Rahman [2000] . . . 34
4.2.4. Bin Yu [2002] . . . 34
4.2.5. Jøsang [2002] (Beta Reputation System) . . . 35
4.2.6. Sabater [2003] (ReGreT) . . . 36
4.2.7. Lik Mui [2003] . . . 37
4.2.8. UniTEC [2003] . . . 38
4.3. Zusammenfassung . . . 38

5. Das generische Vertrauensmodell 41
5.1. Metriken . . . 41
5.1.1. Vertrauensmetrik . . . 41
5.1.2. Gewissheitsmetrik . . . 42
5.1.3. Bewertungsmetrik . . . 42
5.1.4. Wertmetrik . . . 43
5.2. Faktoren . . . 43
5.2.1. Alterungsfaktor . . . 43
5.2.2. Abnutzungsfaktor . . . 44
5.2.3. Aktualisierungsgewicht . . . 45

6. Das generische Vertrauensmodell in UniTEC 47
6.1. Use-Cases . . . 47
6.2. Architektur . . . 49
6.3. Integration lokaler Vertrauensmodelle und -algorithmen . . . 50
6.3.1. Anforderungen . . . 50
6.3.2. Integrationsschritte . . . 53
6.4. Vorteile des generischen Vertrauensmodells . . . 53
6.4.1. Unterschiedliche lokale Vertrauensmodelle . . . 53
6.4.2. Einheitliche globale Metrik . . . 53
6.4.3. Einheitliche lokale Metrik . . . 54

7. Umsetzung vorhandener Vertrauensmodelle und -algorithmen 55
7.1. Allgemein . . . 55
7.1.1. Aktualisierungstyp . . . 55
7.1.2. Metriken . . . 55
7.2. Vertrauensmodelle . . . 56
7.2.1. Jonker [1999] . . . 56
7.2.2. Bin Yu [2000] . . . 56
7.2.3. Rahman [2000] . . . 57
7.2.4. Bin Yu [2002] . . . 58

7.2.5. Jøsang [2002] (Beta Reputation System) 59
7.2.6. Sabater [2003] (ReGreT) 62
7.2.7. UniTEC [2003] 62
7.3. Erkenntnisse bei der Umsetzung 63
7.3.1. Grundeigenschaften 63
7.3.2. Grundfunktionalität 64

8. Analyse **69**
8.1. Testszenarien 69
8.2. Erwartungen 71
8.3. Evaluation und Ergebnisse 71
8.3.1. Bin Yu [2000] 71
8.3.2. Rahman [2000] 73
8.3.3. Bin Yu [2002] 76
8.3.4. Jøsang [2002] (Beta Reputation System) 77
8.3.5. Sabater [2003] (ReGreT) 81
8.3.6. Lik Mui [2003] 83
8.3.7. UniTEC [2003] 83
8.3.8. Allgemeine Ergebnisse 86
8.4. Zusammenfassung der Ergebnisse 90

9. Fazit und Ausblick **93**
9.1. Fazit 93
9.2. Ausblick 93

A. Java Dokumentation **95**
A.1. Kategorienkomponente, Paket `unitec.categories` 95
A.1.1. Basiselemente 95
A.1.2. Manager 96
A.2. Vertrauenskomponente, Paket `unitec.trust` 98
A.2.1. Vertrauensidentifikatoren 99
A.2.2. Vertrauenswerte 100
A.2.3. Vertrauensmanager 101
A.2.4. Vertrauensinitialisierung 106
A.2.5. Vertrauensaktualisierung 106
A.2.6. Vertrauenswertrückgabe 108
A.2.7. Vertrauensalgorithmen 108
A.3. Bewertungskomponente, Paket `unitec.rating` 112
A.3.1. Bewertungswerte 113
A.3.2. Bewertungen 113
A.3.3. Bewertungsmanager 115
A.4. Test Framework, Paket `unitec.tests.trust` 116
A.4.1. Testszenarien 116
A.4.2. Ausgabe 116

A.4.3. Durchführung der Testläufe 118

B. Testergebnisse **119**

C. Glossar **137**

D. Literaturverzeichnis **147**

Stichwortverzeichnis **151**

Erklärung **155**

Abbildungsverzeichnis

2.1. Vertrauensbegriffe 19
2.2. Beispiel zu Spezialisierung von Kategorien 21

3.1. Aktualisierungsgewichte 26

5.1. Vertrauensverlauf mit Abnutzungsfaktor (Alterungsfaktor $\alpha = 0.5$) 45
5.2. Vertrauensverlauf mit Abnutzungsfaktor (Alterungsfaktor $\alpha = 0.2$) 46

6.1. Anwendungsfälle (Use Cases) für Vertrauensberechnungen bei UniTEC 48
6.2. Architektur der Vertrauenskomponente in UniTEC 50

7.1. Vertrauens- und Gewissheitsdynamik bei Bin Yu [2002] im Testszenario RandomRatings (relative Gewichtung) 67
7.2. Vertrauens- und Gewissheitsdynamik bei dem Beta Reputation System im Testszenario RandomRatings (absolute Gewichtung) 67

8.1. Beispiel einer Vertrauensdynamik 69
8.2. Vertrauensdynamik bei Bin Yu [2000] im Testszenario *MaximalThenMinimalRatings* 72
8.3. Vertrauensdynamik bei Bin Yu [2000] im Testszenario *KeepPositive* 73
8.4. Vertrauensdynamik bei Bin Yu [2000] mit gleichstarker Behandlung positiver und negativer Bewertungen ($0 < \alpha < 0.5$ und $-0.5 < \beta < 0$) 74
8.5. Vertrauensdynamik bei Bin Yu [2000] mit stärkeren Einfluss negativer Bewertungen ($0.25 < \alpha < 0.5$ und $-0.75 < \beta < -0.5$) 74
8.6. Vertrauens- und Gewissheitsdynamik bei [Abdul-Rahman und Hailes, 2000] . . 75
8.7. Vertrauens- und Gewissheitsdynamik bei [Yu und Singh, 2002] 77
8.8. Vertrauens- und Gewissheitsdynamik bei dem Beta Reputation System mit Gewichtung $g = 1.0$ und Alterungsfaktor $\alpha = 0.5$ 78
8.9. Vertrauens- und Gewissheitsdynamik bei dem Beta Reputation System im Testszenario *KeepPositive* 79
8.10. Vertrauens- und Gewissheitsdynamik bei dem Beta Reputation System ohne Alterungseffekt 80
8.11. Vertrauens- und Gewissheitsdynamik bei dem Beta Reputation System mit unterschiedlichen Gewichtungen im Testszenario *OnlyMaximalRatings* 80
8.12. Vertrauens- und Gewissheitsdynamik bei ReGreT 82

8.13. Vertrauens- und Gewissheitsdynamik bei dem UniTEC-Algorithmus im Testszenario *RandomRatings* . . . 83
8.14. Vertrauens- und Gewissheitsdynamik bei dem UniTEC-Algorithmus im Testszenario *MaximalThenMinimalRatings* . . . 84
8.15. Vertrauens- und Gewissheitsdynamik bei dem UniTEC-Algorithmus im Testszenario *KeepPositive* . . . 85
8.16. Vertrauensdynamik bei UniTEC-Algorithmus mit einem Abnutzungsfaktor (bei Alterungsfaktor $\alpha = 0.5$) . . . 85
8.17. Vertrauensdynamik bei UniTEC-Algorithmus mit einem Abnutzungsfaktor (bei Alterungsfaktor $\alpha = 0.2$) . . . 86
8.18. Vertrauens- und Gewissheitsverlauf des Beta Reputation System bei den *SpecificRatings* Testszenarien mit Alterungsfaktor $\alpha = 0.0$ . . . 87
8.19. Vertrauensverlauf bei Testszenario *MaximalThenMinimalRatings*, Teil 1 . . . 88
8.20. Vertrauensverlauf bei Testszenario *MaximalThenMinimalRatings*, Teil 2 . . . 88
8.21. Vertrauensverlauf bei Testszenario *SpecificRatingsNormal*, Teil 1 . . . 89
8.22. Vertrauensverlauf bei Testszenario *SpecificRatingsNormal*, Teil 2 . . . 89
8.23. Vertrauensverlauf bei Testszenario *SpecificRatingsReversed*, Teil 1 . . . 90
8.24. Vertrauensverlauf bei Testszenario *SpecificRatingsReversed*, Teil 2 . . . 91

A.1. Basisklassen in `com.hp.unitec.category` . . . 95
A.2. Kategorienverwaltung in `com.hp.unitec.category` . . . 98
A.3. Vertrauenswerte in `unitec.trust` . . . 100
A.4. Vertrauensmanager . . . 102
A.5. Vertrauensereignisse in `unitec.trust` . . . 107
A.6. Implementierung von Vertrauensalgorithmen . . . 109
A.7. Bewertungswerte in `unitec.trust.rating` . . . 113

B.1. Vertrauensverlauf im Testszenario *OnlyMaximalRatings*, Teil 1 . . . 120
B.2. Vertrauensverlauf im Testszenario *OnlyMaximalRatings*, Teil 2 . . . 120
B.3. Vertrauensverlauf im Testszenario *OnlyMinimalRatings*, Teil 1 . . . 121
B.4. Vertrauensverlauf im Testszenario *OnlyMinimalRatings*, Teil 2 . . . 121
B.5. Vertrauensverlauf im Testszenario *MinimalThenMaximalRatings*, Teil 1 . . . 122
B.6. Vertrauensverlauf im Testszenario *MinimalThenMaximalRatings*, Teil 2 . . . 122
B.7. Vertrauensverlauf im Testszenario *MaximalInterruptedByMinimal*, Teil 1 . . . 123
B.8. Vertrauensverlauf im Testszenario *MaximalInterruptedByMinimal*, Teil 2 . . . 123
B.9. Vertrauensverlauf im Testszenario *MinimalInterruptedByMaximal*, Teil 1 . . . 124
B.10. Vertrauensverlauf im Testszenario *MinimalInterruptedByMaximal*, Teil 2 . . . 124
B.11. Vertrauensverlauf im Testszenario *SlightlyNegativeThenPositiveRatings*, Teil 1 . 125
B.12. Vertrauensverlauf im Testszenario *SlightlyNegativeThenPositiveRatings*, Teil 2 . 125
B.13. Vertrauensverlauf im Testszenario *SlightlyPositiveThenNegativeRatings*, Teil 1 . 126
B.14. Vertrauensverlauf im Testszenario *SlightlyPositiveThenNegativeRatings*, Teil 2 . 126
B.15. Vertrauensverlauf im Testszenario *SpecificRatingsNormal*, Teil 1 . . . 127
B.16. Vertrauensverlauf im Testszenario *SpecificRatingsNormal*, Teil 2 . . . 127
B.17. Vertrauensverlauf im Testszenario *SpecificRatingsReversed*, Teil 1 . . . 128

B.18. Vertrauensverlauf im Testszenario *SpecificRatingsReversed*, Teil 2 128
B.19. Vertrauensverlauf im Testszenario *SpecificRatingsOrderedAsc*, Teil 1 129
B.20. Vertrauensverlauf im Testszenario *SpecificRatingsOrderedAsc*, Teil 2 129
B.21. Vertrauensverlauf im Testszenario *SpecificRatingsOrderedDesc*, Teil 1 130
B.22. Vertrauensverlauf im Testszenario *SpecificRatingsOrderedDesc*, Teil 2 130
B.23. Vertrauensverlauf im Testszenario *OscilatingRatingsFirstMaximal*, Teil 1 . . . 131
B.24. Vertrauensverlauf im Testszenario *OscilatingRatingsFirstMaximal*, Teil 2 . . . 131
B.25. Vertrauensverlauf im Testszenario *OscilatingRatingsFirstMinimal*, Teil 1 132
B.26. Vertrauensverlauf im Testszenario *OscilatingRatingsFirstMinimal*, Teil 2 132
B.27. Vertrauensverlauf im Testszenario *RandomRatings*, Teil 1 133
B.28. Vertrauensverlauf im Testszenario *RandomRatings*, Teil 2 133
B.29. Vertrauensverlauf im Testszenario *KeepPositive*, Teil 1 134
B.30. Vertrauensverlauf im Testszenario *KeepPositive*, Teil 2 135
B.31. Vertrauensverlauf im Testszenario *KeepPositive*, Teil 3 135

C.1. Beispiel zur Erläuterung einiger Begriffe 140

Tabellenverzeichnis

4.1. Berechnungsvorschrift für T_i laut [Yu und Singh, 2000] 33
4.2. Unsicherheitswert laut [Abdul-Rahman und Hailes, 2000] 34
4.3. Eigenschaften der Vertrauensmodelle aus der Literatur 39
4.4. Bewertungsmetriken der Vertrauensmodelle aus der Literatur 40
4.5. Vertrauensmetriken der Vertrauensmodelle aus der Literatur 40

7.1. Beispiel zur Konvertierung der Bewertungsmetrik von [Yu und Singh, 2000] . . 57
7.2. Umsetzung der Unsicherheitswerte in das generische Vertrauensmodell 58
7.3. Grundeigenschaften der Vertrauensalgorithmen 64

1. Einführung

1.1. Motivation

Täglich vertrauen wir Sachen und Menschen, um die Komplexität von Entscheidungen zu vereinfachen oder überhaupt Entscheidungen zu ermöglichen. Diese Vereinfachung basiert auf ein erhöhtes Risiko, welches bei der Entscheidungstreffung in Kauf genommen wird. Solche Vertrauensentscheidungen beruhen auf einer subjektiven Einschätzung, wie „gut“ dem betrachteten Objekt vertraut werden kann. Dieses Vertrauen kann von Mensch zu Mensch unterschiedlich die Entscheidungsfällung beeinflussen.

Vertrauens-modellierung in Rechnersystemen

Ein interessantes Forschungsgebiet im Bereich der Informatik ist es, diesen Vertrauensbegriff rechnerkonform zu modellieren, so dass ein Computersystem Vertrauensentscheidungen stellvertretend für einen Menschen treffen kann. In einem solchen Modell wird die Art und Weise festgelegt, wie unterschiedliche Vertrauensverhältnisse (z.B. „ich habe absolutes Vertrauen in Objekt x“) beschrieben werden. Ein Vertrauensverhältnis zu einem Objekt soll zudem durch bestimmte Ereignisse beeinflusst werden können (z.B. nach neuen Erfahrungen, Beobachtungen oder Informationen zu dem Objekt). Diese Vorgänge verarbeiten Menschen subjektiv und sind mathematisch schwer beschreibbar. Das Modell muss demzufolge das menschliche Vertrauenskonzept zur Verarbeitung vereinfachen, während es für den Benutzer trotzdem nachvollziehbar bleibt.

UniTEC

UniTEC ist ein verteiltes Reputationssystem, in welchem Empfehlungen zu Objekten geschrieben werden können. Diese Empfehlungen können abgefragt und durch die lesenden Benutzer bewertet werden. Agenten interagieren stellvertretend für dessen Benutzer in dieser „Empfehlungs-Gesellschaft“ miteinander. Grundlagen zu UniTEC ist die Arbeit in [Kinateder und Rothermel, 2003].

Es ist nun Aufgabe der beteiligten Agenten im UniTEC System durch Kommunikation und durch die Eingaben der Benutzer (z.B. die Bewertungen von Empfehlungen Dritter) die Vertrauensbeziehungen zu erlernen. Die Dynamik der Vertrauensverhältnisse in dieser virtuellen Gesellschaft soll die realen Vertrauensverhältnisse zwischen den Benutzer, die hinter den Agenten stehen, so gut wie möglich widerspiegeln.

Vertrauens-aktualisierungs-algorithmen

Die Dynamik des Zuwachses und Nachlasses des Vertrauens wird durch Vertrauensaktualisierungsalgorithmen beschrieben. In der Literatur wurden bereits unterschiedliche Vertrauensaktualisierungsalgorithmen vorgestellt, welche jeweils auf unterschiedliche Vertrauensmodelle beruhen. In dieser Arbeit wird später detailliert auf die einzelnen Kandidaten eingegangen.

Ziel der Arbeit ist es ein *generisches Vertrauensmodell* zu konzipieren, welches eine Ober-

menge an Modellierungsmöglichkeiten anbietet, die in den einzelnen Vertrauensmodellen der Literatur berücksichtigt werden. Eine Integration des generischen Vertrauensmodells in UniTEC soll es ermöglichen, eine möglichst grosse Anzahl an Vertrauensmodellen der Literatur in UniTEC einsetzen zu können, um somit die unterschiedlichen Vertrauensaktualisierungsalgorithmen erstmalig vergleichen zu können. Das generische Vertrauensmodell bietet zudem eine einheitliche Metrik zur Kommunikation von Vertrauen zwischen Agenten, die unterschiedliche *lokale Vertrauensmodelle* einsetzen wollen.

1.2. Struktur der Arbeit

In Kapitel 2 wird zunächst auf die Begrifflichkeit, die zu Vertrauen existiert, eingegangen. Wichtige Erkenntnisse aus der Literatur zu dem Thema werden durch Definitionen ergänzt, die in der restlichen Arbeit verwendet werden. Nachfolgend werden in Kapitel 3 relevante Bestandteile von Vertrauensmodellen diskutiert. Wichtige Vertrauensmodelle aus der Literatur werden in Kapitel 4 klassifiziert und beschrieben. Aufgrund der Bestandteile eines Vertrauensmodells und den Erkenntnissen aus der Literatur wird das generische Vertrauensmodell im Kapitel 5 festgelegt. Die Umsetzung des generischen Vertrauensmodells im UniTEC-System, das Zusammenspiel der diversen Komponenten und die Möglichkeiten zur Integration weiteren Vertrauensmodellen in das System werden in 6 erörtert. In Kapitel 7 werden die technischen Details zur Umsetzung jeder der betrachteten Vertrauensmodelle und -algorithmen besprochen. Letztendlich werden die realisierten Vertrauensalgorithmen in Kapitel 8 miteinander verglichen und einer detaillierten Analyse unterzogen. Schließlich werden in Kapitel 9 ein Fazit der Erkenntnisse und ein Ausblick auf weitere Forschungsmöglichkeiten auf diesem Gebiet gegeben.

2. Das Konzept „Vertrauen“

2.1. Vertrauensbegriff

Die unterschiedlichsten Fachgebiete beschäftigten sich bereits mit dem Thema Vertrauen und Reputation: Soziologie, Psychologie, Ökonomie und Spieletheorie. Neuerdings auch die Informatik, allerdings im Zusammenhang mit einer oder mehreren der anderen Disziplinen. In der Literatur zu dem Thema wird oft die Vertrauensdefinition von [Gambetta, 2000] erwähnt: Gambetta

> Vertrauen (oder, symmetrisch dazu Misstrauen) ist ein bestimmtes Niveau an subjektiver Wahrscheinlichkeit, mit dem ein Agent einschätzt, ob ein anderer Agent oder Gruppe von Agenten eine bestimmte Aktion durchführen wird, jedoch bevor er die Aktion verfolgen kann (unabhängig von seinen Möglichkeiten die Aktion überhaupt jemals verfolgen zu können) und in einem Kontext in dem es auch seine Aktionen beeinflusst. (*aus dem Englischen übersetzt*)

Lik Mui hingegen ersetzt den Begriff „Wahrscheinlichkeit“ mit der allgemeineren Aussage der „Erwartung“ und definiert somit Vertrauen als „eine subjektive Erwartung, die ein Agent über das zukünftige Verhalten eines anderen Agenten hat, bezogen auf die Historie von Begegnungen zwischen den Agenten.“ (vgl. Mui [2003]). Lik Mui

Ähnlich wird in [Abdul-Rahman und Hailes, 2000] der Begriff Reputation definiert: „Reputation ist eine Erwartung über das Verhalten eines Agenten, basierend auf Informationen über oder Beobachtungen seines Verhaltens in der Vergangenheit.“. Der Unterschied zu der Vertrauensdefinition von Lik Mui ist, dass dort explizit eine Beziehung zwischen zwei Agenten beschrieben wird, während bei Abdul-Rahman und Hailes keine genaue Angabe zu dem Aktor der Reputation gemacht wird. Abdul-Rahman

Hier verallgemeinern wir die Definition von Lik Mui und setzen keine Begegnungen voraus, sondern verwenden wie bei Abdul-Rahman und Hailes lediglich „Informationen“ oder „Beobachtungen“ als Basis zur Vertrauensaussage (wobei diese durchaus aus Begegnungen entstanden sein können).

Definition 1 *Vertrauen ist eine subjektive Einschätzung eines Aktors, ob das betrachtete Objekt in einem bestimmten Kontext, zu einem bestimmten Ziel und zu einem bestimmten Zeitpunkt mit ihm kooperieren wird, basierend auf Informationen über oder Beobachtungen des Verhaltens des betrachteten Objektes in der Vergangenheit.* Vertrauen

Die zwei Beteiligten (der Aktor und das betrachtete Objekt) stehen somit in einem Vertrauensverhältnis zueinander:

Vertrauensverhältnis

Definition 2 *Ein Vertrauensverhältnis beschreibt das Vertrauen, welches ein* Aktor *in einem* Vertrauten *in einem bestimmten Kontext zu einem bestimmten Ziel an einem bestimmten Zeitpunkt hat.*

Vertrauen wird in der Definition 1 als *Maß von Vertrauen* verwendet, und stellt somit keine positive oder negative Einschätzung dar, wie es bei Gambetta der Fall ist, wo deswegen explizit Vertrauen und Misstrauen erwähnt werden. Oft wird der Begriff „Vertrauen“ alleine im positiven Sinne verwendet, wobei Misstrauen das negative Gegenstück repräsentiert. In dieser Arbeit wird das positive Vertrauen als solches explizit kennzeichnen („großes Vertrauen“, „volles Vertrauen“, „positives Vertrauen“).

Kontext

Der *Kontext* als weitere Dimension des Vertrauens ermöglicht es unterschiedliche Vertrauensverhältnisse zu einem Agenten in unterschiedlichen Bereichen oder Situationen zu bilden: Jemand kann großes Vertrauen in einen Automechaniker haben, dass er sein Auto reparieren wird (vgl. Jøsang u. a. [2003a]), jedoch in Computerfragen (anderer Kontext) herrscht zwischen den beiden ein komplett anderes Vertrauensverhältnis.

Aus dem Beispiel wird auch klar, dass das Vertrauensverhältnis nicht unbedingt auf Gegenseitigkeit beruht: das Vertrauen von A in B im Kontext K muss nicht unbedingt gleich dem Vertrauen von B in A im gleichen Kontext K sein.

Vertrauensziel

Das *Ziel* einer Vertrauensaussage ist eine weitere Dimension, die beschreibt, wofür eine Vertrauensaussage in dem bestimmten Kontext verwendet wird. Im UniTEC-System gibt es zwei möglichen Vertrauensziele: a) dass ein fremdes Pseudonym eine nützliche Empfehlung abgibt und b) dass ein fremdes Pseudonym eine Anfrage so vermittelt, dass daraus nützliche Empfehlungen zurückkommen. Weitere Ziele werden im UniTEC-System zunächst nicht betrachtet, sind aber durchaus denkbar (z.B. dass jemand ein Thema didaktisch erklären kann, dass jemand in einem Kontext unehrliche Fremde erkennen kann, usw.).

Zeit

Die *zeitliche Dimension* in der Vertrauensaussage ermöglicht eine Änderung der Vertrauensaussagen im Laufe der Zeit. Um einzuschätzen, ob eine Kooperation mit einem Partner wahrscheinlich ist, werden die Vertrauensverhältnisse evaluiert. Jedoch erst nach dem Geschehen der kooperativen Aktion kann festgestellt werden, ob diese Einschätzung gut (falls die Kooperation erfolgreich war) oder schlecht (im Falle eines unkooperativen Partners) war. Nach dieser Beurteilung des Ausganges der Aktion kann sich das Vertrauensverhältnis zwischen den Agenten gegebenenfalls ändern. Ist die Beurteilung positiv, so kann sich das Vertrauen in diesem Kontext zu dieser Person z.B. erhöhen, ist sie negativ, so kann sich das Vertrauen verringern. Es entsteht somit eine *zeitliche Dynamik des Vertrauens* (mehr dazu in Abschnitt 2.4).

Im Folgenden werden oft die Begriffe *Bewerter* und *Empfehler* anstatt Vertrauensaktor bzw. vertrautes Objekt verwendet. Diese sind die beteiligten Personen in einer Vertrauensbeziehung im UniTEC-System.

2.1.1. Vertrauensabsicht

Das Konzept von Vertrauen wird benötigt, um die Komplexität von Entscheidungen zu verringern, indem ein gewisses Maß an Unsicherheit in eine Abhängigkeit zu dem vertrautem Objekt umgewandelt wird. Damit ist diese Entscheidung mit einem bestimmten Risiko verbunden. Dieses Konzept der Abhängigkeit wird in [McKnight und Chervany, 1996] bei der Definition der Vertrauensabsicht (*trusting intention*) deutlich:

> Die Vertrauensabsicht ist der Ausmaß an Abhängigkeit, den eine Person gegenüber einem anderen Beteiligten in einer bestimmten Situation mit einem Gefühl einer relativen Sicherheit bereit ist einzugehen, obwohl negative Folgen möglich sind. (*aus dem Englischen übersetzt*)

Vertrauensbegriffe laut McKnight

McKnight und Chervany erläutern zusätzlich zu der Vertrauensabsicht weitere fünf Vertrauensbegriffe, die unterschiedliche Ausprägungen von Vertrauen beschreiben. Somit wird für die Forschungsarbeiten in diesem Bereich eine einheitliche Begriffswelt geschaffen:

Vertrauensglauben (*trusting beliefs*) deutet auf die kognitiven Vorstellungen, die eine Person gegenüber einer anderen Person hat. Auf diese Überzeugungen basieren letztendlich die Vertrauensabsichten. Aus der Vertrauensabsicht folgt nun das Vertrauensverhalten (*trusting behaviour*): Nun wird die Absicht konkret in eine Handlung umgesetzt, es kommt konkret zu der Abhängigkeit. Das Systemvertrauen (*system trust*) beschreibt die Einschätzung einer Person zu gewissen Strukturen im System (z.B. Verträge, Gesetze, usw.). Es ändert nichts an den persönlichen Vertrauensglauben, es kann jedoch die Vertrauensabsichten beeinflussen (und damit auch das Verhalten). Unter veranlagertes Vertrauen (*dispositional trust*) verstehen McKnight und Chervany die Grundvertrauenseinstellung einer Person, unabhängig von der Situation oder vertraute Person. Zuletzt beschreibt die situationsbezogene Entscheidung zu Vertrauen (*situational decision to trust*) das Ausmaß an Abhängigkeit, das eine Person grundsätzlich in einer bestimmten Situation maximal eingeht, unabhängig des anderen Beteiligten.

In dieser Arbeit wird als Vertrauen die Vertrauensabsichten verstanden, die in konkreten Situationen gegenüber anderen Personen zustande kommen.

2.1.2. Reputation

In einigen Arbeiten (Abdul-Rahman und Hailes [2000], Dellarocas [2003b]) wird der Begriff Reputation anstelle von Vertrauen verwendet, oder es wird keine klare Abgrenzung zwischen Vertrauen und Reputation gemacht.

Die Vertrauensdefinition 1 beschreibt konkret die Einschätzung eines Aktors gegenüber einem betrachteten Objekt. In dieser Arbeit ergibt sich die Reputation aus der gesammelten Menge an Vertrauensverhältnisse zwischen einer gewissen Anzahl von Aktoren gegenüber diesem Objekt.

Reputation

Definition 3 *Reputation ist eine zusammengefasste Vertrauensaussage einer Menge von Aktoren gegenüber einem Objekt.*

Eine globale Reputation wäre eine Zusammenfassung sämtlicher Vertrauensaussagen aller Aktoren, die ein Vertrauensverhältnis zu einem Objekt haben. Diese ist verständlicherweise nicht durch einzelne Aktoren ermittelbar: Selbst wenn ein Aktor das Vertrauensverhältnis sämtlicher Beteiligten einholen könnte, wären diese ermittelten Daten nur in dem Maße glaubwürdig, als auch der jeweilig befragte Beteiligte vertrauenswürdig ist. Trotzdem kann eine *relative Reputation* ermittelt werden, indem die Aussagen Dritter gewichtet (mit dem entsprechenden Vertrauen) zusammengefasst werden. Diese Art von Reputation wird in einigen Arbeiten durch unterschiedlichste Algorithmen berücksichtigt (Yu und Singh [2000], Abdul-Rahman und Hailes [2000], Sabater [2003], Mui [2003]).

Reputationsermittlung

2.2. Kooperation

In der Definition zu Vertrauen (Definition 1) wird zuletzt *Kooperation* als Motivation des Vertrauens erwähnt. Im Sinne von UniTEC kann eine Kooperation z.B. bedeuten, eine Empfehlung zu einem Objekt abzugeben, die mit der späteren Meinung des Anfragenden zu dem Objekt übereinstimmt, so dass der Anfragende durch Evaluierung der Empfehlungen seine Entscheidung treffen kann (z.B. zum Kauf des Objektes).

Kooperation in UniTEC

Die erwartete Kooperation hängt direkt mit dem Ziel der Vertrauensaussage zusammen: Es wird eine Kooperation zu dem bestimmten Ziel erwartet, das auch das Ziel der Vertrauensaussage war. Wenn also positives Vertrauen mit dem Ziel „gute Vermittlung der Anfrage“ gesetzt wurde, wird als kooperative Gegenleistung eine „gute Vermittlung der Anfrage“ erwartet und sonst nichts.

Vertrauensziel und Kooperation

Als Beispiel wie unterschiedliche Vertrauensziele unterschiedliche Vertrauensverhältnisse hervorrufen können, gelte eine Person *A*, die zu einem Kontext keine Expertise hat, jedoch viele weitere Personen kennt, die in diesem Gebiet große Expertise haben. Somit sollte sich das Vertrauen in Person *A* mit dem Ziel „gute Vermittlung der Anfrage“ nach einer Anzahl positiver Erfahrungen in diesem Kontext zu diesem Ziel auf ein hohes Niveau einpendeln, während das Vertrauen im selben Kontext mit dem Ziel „gute Empfehlungen“ nicht so hoch sein muss.

Durch den Dynamismus der Vertrauensverhältnisse werden sich nach vielen Interaktionen zwischen den Agenten zu jedem Kontext Benutzermengen bilden, die starkes Vertrauen zueinander aufweisen. Laut Untersuchungen in [Ziegler und Lausen, 2004] befinden sich in einer solchen Benutzermenge Benutzer ähnlicher Gesinnung (in diesen bestimmten Kontext). Somit bekommt jeder Benutzer nach einer gewissen Anzahl von Erfahrungen Empfehlungen, die eher der eigenen Meinung entsprechen, da sich in seiner Nachbarschaft Benutzer mit ähnlicher Meinung befinden.

Ähnlichkeitsgruppen

2.3. Vertrauenszustände

Das Vertrauensverhältnis kann durch unterschiedliche Vertrauensbegriffe beschrieben werden. Folgende Begriffe verwenden wir, um bestimmte Zustände des Vertrauens zu beschreiben:

- *Absolutes Misstrauen*: Es wird nie eine Kooperation vom betrachtetem Objekt erwartet.
- *Misstrauen*: Es wird eher keine Kooperation erwartet. Hier sind alle unterschiedlichen Grade an Misstrauen zwischen dem absoluten Misstrauen und dem neutralen Vertrauensverhältnis angesiedelt.
- *Neutrales Vertrauensverhältnis*: Hier wird dem betrachteten Objekt weder vertraut noch misstraut.
- *Positives Vertrauen*: Es wird eine Kooperation erwartet. Alle möglichen Abstufungen an positiven Vertrauen zwischen dem neutralen Vertrauensverhältnis und dem blinden Vertrauen sind hier zu finden.
- *Blindes Vertrauen*: Dieser Begriff beschreibt eigentlich kein Vertrauensverhältnis in dem Sinne, als dass der bewertende Agent selbst eine Entscheidung nach Erwägung der Vertrauensbeziehung fällen würde: Hier übernimmt der fremde Agent die Entscheidungsrolle. Deswegen wird dieser Wert in einigen Forschungsarbeiten zu dem Thema nicht im Bereich von „Vertrauen“ angesiedelt (vgl. Marsh [1994]).

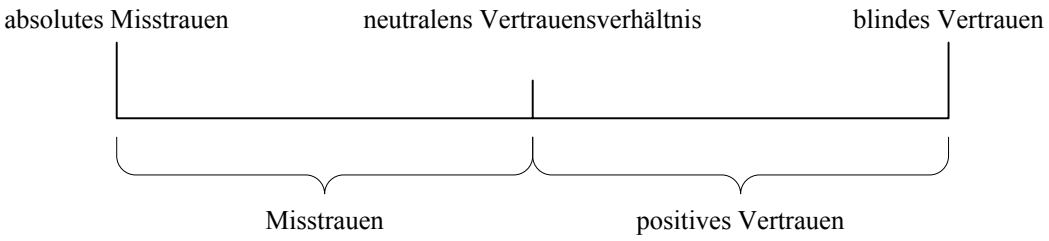


Abbildung 2.1.: Vertrauensbegriffe

2.4. Vertrauensdynamik

Bei einer Begegnung mit einem neuen fremden Objekt wird ein *initiales Vertrauensverhältnis* festgelegt. Dieses Vertrauen beschreibt die Grundveranlagung einer Person in einem bestimmten Kontext zu vertrauen noch bevor irgendwelche Erfahrungen mit dem Objekt gemacht wurden. Nachfolgende Erfahrungen oder Beobachtungen zu dem fremden Objekt verursachen Änderungen in diesem Vertrauensverhältnis. In [Jonker und Treur, 1999] werden einige mögliche Dynamiken des Vertrauensverhältnisses im Laufe der Zeit vorgestellt, die hier etwas anders angeordnet und erweitert werden:

Initiales Vertrauensverhältnis

Statisches Vertrauensverhältnis Jedem fremden Pseudonym wird stets ein bestimmtes Vertrauen zugesprochen. Extremfälle davon sind:

1. *Stets blindes Vertrauen*: Jedem betrachtetem Objekt wird stets das blinde Vertrauen zugesprochen.

2. *Stets absolutes Misstrauen*: Jedem betrachtetem Objekt wird stets das absolute Misstrauen zugesprochen.

Dynamisches Vertrauensverhältnis Wesentlich einleuchtender ist es, das Vertrauensverhältnis dynamisch im Laufe der Zeit abzuändern. Folgende Charakteristiken können dabei festgestellt werden:

1. *Mit oder ohne „Fangzustände“*: Ein Fangzustand ist ein Zustand, den das Vertrauensverhältnis erreichen kann und wonach es sich nicht mehr ändert. Dieser Zustand kann z.B. nach einer bestimmten Anzahl oder eine Sequenz von negativen bzw. positiven Ereignissen erreicht werden. Der Fangzustand kann das Vertrauensverhältnis z.B. im blinden Vertrauen oder im absoluten Misstrauen einfangen. Ohne einen Fangzustand kann sich das Vertrauensverhältnis jederzeit durch eine geeignete Anzahl von Ereignissen wieder ändern.

2. *Symmetrie der Dynamik*

   a) *Symmetrische Dynamik*: Hier ist eine gleichschnelle Änderung des Vertrauensverhältnisses sowohl bei positiven als auch bei negativen Erfahrungen erkennbar.

   b) *Asymmetrische Dynamik*

      i. *Langsame positive und schnelle negative Dynamik*: Es sind viele positive Erfahrungen notwendig um das Vertrauen aufzubauen, nur wenige negative reichen aus, um es abzubauen.

      ii. *Schnelle positive und langsame negative Dynamik*: Es sind wenige positive Erfahrungen notwendig um das Vertrauen aufzubauen, jedoch viele negative, um es wieder abzubauen.

2.5. Kontexte

Wie bereits erwähnt, sind unterschiedliche Kontexte für Vertrauensaussagen in vielen Anwendungen durchaus wichtig. Ein Kontext repräsentiert einen bestimmten Wissensbereich. In UniTEC werden die unterschiedlichen Kontexte in Form von Kategorien modelliert (vgl. Kinateder und Rothermel [2003]).

Zwei Kategorien können auf unterschiedliche Arten Beziehungen zueinander aufweisen. Eine Kategorie kann eine Spezialisierung einer anderer sein: z.B. die Kategorien *Hauseinrichtung*, *Möbel*, *Küchenmöbel* und *Küchenregale* weisen einen zunehmenden Spezialisierungsgrad auf (siehe Abbildung 2.2). Weitere Beziehungen zwischen Kategorien können sein: Ganzes-Teile-Beziehung (z.B. *PKW* und *PKW-Motor*), semantische Ähnlichkeit (z.B. *Sonnenbrillen* und *Sonnenschirme*), gemeinsame Klasse (z.B. Klasse von Automobilherstellern mit *Mercedes-Benz* und *BMW*), usw.

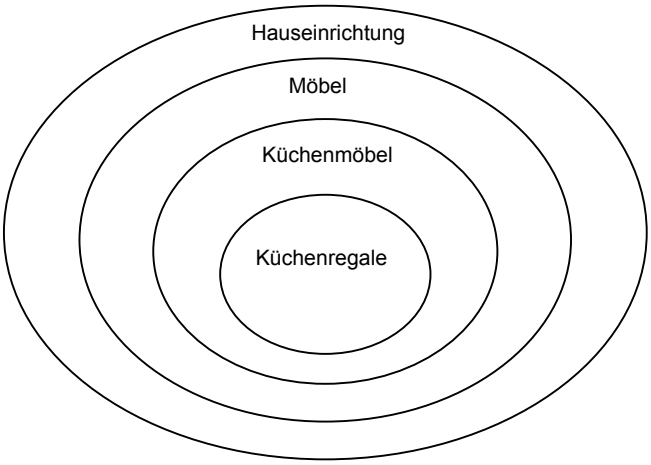


Abbildung 2.2.: Beispiel zu Spezialisierung von Kategorien

Die für UniTEC eingeführte *semantische Verwandtschaft* zwischen zwei Kategorien beschreibt auf einfache Weise die *Stärke* der Beziehung zwischen den Kategorien. Diese beschreibt nicht konkret die *Art* der Beziehung. Somit kann die semantische Verwandtschaft im UniTEC in einem einfachen gewichteten Graphen dargestellt werden (siehe dazu Kinateder und Rothermel [2003]).

semantische Verwandtschaft zwischen Kategorien

2.6. Verhaltensmuster

Ein Empfehler kann unterschiedliche Verhaltensmuster aufweisen, die die erzeugten Vertrauensdynamiken und die Entscheidungen der Vertrauensaktualisierungsalgorithmen unterschiedlich beeinflussen können:

Grad der Wechselhaftigkeit Der Ähnlichkeitsgrad zwischen dem Empfehler und den Bewertenden spielt hier eine Rolle. Personen mit ähnlicher Gesinnung weisen einen hohen Ähnlichkeitsgrad auf, Personen mit großer Meinungsverschiedenheit weisen einen geringen Ähnlichkeitsgrad auf. Der Ähnlichkeitsgrad bezieht sich jeweils auf einen bestimmten Kontext zu einem bestimmten Ziel und zu einem bestimmten Zeitpunkt.

Ähnlichkeitsgrad

- *Stetiges Verhalten*: Der Empfehler ändert seinen Ähnlichkeitsgrad zu dem Anfragenden in dem bestimmten Kontext im Laufe der Zeit nicht oder nur geringfügig. Wenn der Ähnlichkeitsgrad für das Ziel „nützliche Empfehlung abgeben“ z.B. groß ist, werden stets nützliche Empfehlungen abgegeben. Bei geringer Ähnlichkeit bekommt der Anfragende stets untaugliche Empfehlungen. Die Bandbreite des Ähnlichkeitgrades im Laufe der Zeit ist somit in der betrachteten Zeitspanne unter einem bestimmten Schwellwert. In diesem Fall ist eine gute Vorhersage der Kooperationsbereitschaft des Empfehlers möglich.

- *Wechselhaftes Verhalten*: Der Empfehler weist einen wechselhaften Ähnlichkeitsgrad zu dem Anfragenden auf, d.h. z.B. manchmal werden nützliche, manchmal untaugliche Empfehlungen abgegeben. Die Bandbreite des Ähnlichkeitgrades im betrachteten Zeitraum im betrachteten Kontext geht hier über einen gewissen Schwellwert. Die Größenordnung dieser Bandbreite beschreibt zugleich das Ausmaß der Wechselhaftigkeit. Grund für ein wechselhaftes Verhalten kann die Betrachtung eines zu generischen Kontextes, oder auch tatsächlich unterschiedliche Meinungen im selben Kontext sein. Der Spezialisierungsgrad einer Kategorie (siehe Abschnitt 2.5) ist nicht direkt aus dem Abhängigkeitsgraphen ersichtlich, was das Erkennen des Grundes der Wechselhaftigkeit erschwert. Spätere Arbeiten könnten hier versuchen die Empfehlungen stets in den spezialisiertesten Kategorien anzusiedeln.

Ehrlichkeit

- *Ehrliches Verhalten*: Eine Empfehlerin ist dann ehrlich, wenn sie versucht durch ihre Empfehlungen kein anderes Ziel zu erreichen als eine faire Kooperation im Sinne von UniTEC.
- *Unehrliches Verhalten*: Ein Empfehler ist dann unehrlich, wenn er versucht durch seine Empfehlungen andere Ziele als die Kooperation zu erreichen, wie z.B. den kurzfristigen Gewinn an Reputation, der anschließend durch unfaire Empfehlungen ausgenutzt werden kann.

Verhalten des Bewerters

Auf das Verhalten des Bewerters wird hier nicht eingegangen, da davon ausgegangen wird, dass er sich stetig und ehrlich verhält. Dies kann durch die Tatsache begründet werden, dass seine Bewertungen lediglich lokale Bedeutung haben und er keine Vorteile durch unehrliche Bewertungen erlangt. Die Bewertungen mit lokaler Bedeutung reflektieren somit die reelle Meinung des Bewertenden. Falls die Bewertungen an fremde Pseudonyme weitergegeben werden, handelt der Aktor dieser Aktion als Empfehler, und es gelten die obigen Verhaltensmuster.

3. Bestandteile von Vertrauensmodellen und -algorithmen

Unterschiedliche Vertrauensmodelle zum Einsatz in Computersystemen wurden in der Vergangenheit untersucht, zum Teil mit unterschiedlichen Semantiken und Ansichten, was unter Vertrauen zu verstehen ist (wie in [McKnight und Chervany, 1996] ausführlich beschrieben, siehe auch Abschnitt 2.1.1). Ein Vertrauensmodell beschreibt eine Reihe von Eigenschaften, Berechnungsvorschriften und Metriken, um mit Vertrauensinformationen umzugehen.

Im Folgenden werden die wichtigen Aspekte eines Vertrauensmodells diskutiert.

3.1. Vertrauensmetrik

Die wichtigste Komponente eines Vertrauensverhältnisses ist eine Metrik, die eine Vertrauensbeziehung quantifiziert.

Hiermit können die unterschiedlichen Vertrauenszustände aus Abschnitt 2.3 beschrieben werden. Die Metrik enthält meistens ein Minimum, das für ein *absolutes Misstrauen* steht und ein Supremum, welches das *blinde Vertrauen* repräsentiert. Zusätzlich kann ein *neutraler Vertrauenswert* definiert sein, der einen neutralen Vertrauensgrad darstellt (weder positives Vertrauen noch Misstrauen). Werte über diesem neutralen Vertrauenswert stellen unterschiedliche Grade an positiven Vertrauen dar, darunter liegen Werte, die ein Misstrauen darstellen.

In einem Vertrauensmodell kann die Vertrauensmetrik auf unterschiedlichste Weise definiert sein:

- *Skalare Metrik* mit eindeutigen Minimum und Maximum, z.B. der Wertebereich $[0..1]$. Der Minimalwert (hier 0) repräsentiert das absolute Misstrauen und der Maximalwert (hier 1) entspricht dem blinden Vertrauen. Die genaue Semantik der Zwischenwerte ist jedoch sehr subjektiv. Eine solche Metrik lehnt sich stark an das Bayesche Wahrscheinlichkeitskonzept. Die dortigen Erkenntnisse der subjektiven (und persönlichen) Interpretation einer Wahrscheinlichkeit können hier übernommen werden. In [Winkler, 1972] werden dazu einige Möglichkeiten detailliert erläutert, unter anderem durch Gewinnspiele und Wettchancen.

- *Sprachliche Konstrukte*, die nach dem Vertrauensgrad geordnet werden können: z.B. absolutes Misstrauen, mittleres Misstrauen, neutrales Vertrauen, mittleres positives Vertrauen

und blindes Vertrauen. Die einzelnen Begriffe sind zwar eindeutig festgelegt, jedoch ist deren Semantik unscharf und subjektiv.

- *Diskrete Metrik*, wo eine fest definierte Anzahl von Vertrauensgrade festgelegt ist. Die einzelnen Vertrauensgrade entsprechen meistens auch sprachlichen Konstrukten.
- *Binäre Metrik* als besondere Form der diskreten Metrik, wo nur zwei mögliche Zustände existieren: Misstrauen und Vertrauen.

3.2. Vertrauensgewissheit

Zu jedem Vertrauensverhältnis gehört auch eine Gewissheit. Sie beschreibt, mit welcher Sicherheit der Wert der Vertrauensmetrik korrekt ist. In dieser Metrik repräsentiert ein Minimum die *komplette Ungewissheit* und das Supremum die *totale Sicherheit*. Mögliche Beschreibungsvarianten für eine Gewissheit sind:

- *Skalare Metrik* mit eindeutigen Minimum und Maximum, z.B. ein beliebiger Wert im Bereich $[0..1]$. Diese Metrik wird meist als Wahrscheinlichkeitswert verwendet (z.B. „mit 60%iger Gewissheit“)
- *Diskrete Metrik*, mit einer fest definierten Anzahl von Gewissheitsgraden. Die einzelnen Gewissheitsgrade entsprechen meistens auch sprachlichen Konstrukten, wie z.B. „komplett unsicher“, „total sicher“.
- *Binäre Metrik* als besondere Form der diskreten Metrik, wo nur zwei mögliche Zustände existieren: Sicher oder unsicher.

Gewissheit und Vertrauenswert

Die Gewissheit kann auf unterschiedliche Weise in die Betrachtungen eines Vertrauensverhältnisses einfließen. Es kann eine mögliche Toleranzspanne in der Vertrauensmetrik angeben. Bei einer totalen Gewissheit würde diese Spanne minimal klein sein, bei der kompletten Ungewissheit würde die Spanne die komplette Bandbreite der Vertrauensmetrik ausschöpfen. Bei der kompletten Ungewissheit spielt somit der Wert aus der Vertrauensmetrik keine Rolle mehr, da das Vertrauen einen beliebigen unbekannten Wert aus der kompletten Vertrauensmetrik annehmen könnte.

3.3. Vertrauensaktualisierungsereignisse

Mit den festgelegten Metriken lassen sich nun die Vertrauensverhältnisse beschreiben. Änderungen bei diesen Vertrauensverhältnissen werden durch Vertrauensaktualisierungsereignisse hervorgerufen. Die möglichen Ereignisse werden durch das Vertrauensmodell festgelegt und beschrieben. Die gängigste Art eines solches Ereignisses ist eine Bewertung einer Begegnung.

Zwei Beteiligte interagieren miteinander, es kommt zu einer Kooperation (oder nicht), schließlich wird diese Interaktion bewertet. Im UniTEC-System ist eine Interaktion durch dem Empfang und Beurteilung einer Empfehlung gegeben. Der Benutzer kann anschließend diese Empfehlung bewerten, und somit sein Vertrauensverhältnis zu dem Urheber der Empfehlung beeinflussen.

Mögliche Ereignisse, die das Vertrauen verändern können:

- Neue Erfahrungen mit dem Empfehler (in Form einer Bewertung ausgedrückt).
- Der Empfang von Zeugenaussagen (z.B. wenn Dritte die Information ihres Vertrauensverhältnis zu einem betrachtetem Objekt weitergeben)
- Das Vorübergehen der Zeit kann ebenfalls zu einer Änderung des Vertrauens beitragen, auch wenn keine neue Erfahrungen oder Beobachtungen gemacht werden.

3.4. Gewichtung einer Vertrauensaktualisierung

Jedes Ereignis, das eine Aktualisierung des Vertrauens hervorruft (siehe Abschnitt 3.3), kann unterschiedlich gewichtet werden. Damit wird die Bedeutung dieser Aktualisierung im Vergleich zu anderen Aktualisierungen beschrieben.

Als Aktualisierungsgewicht sind folgende Abstufungen möglich:

- *Bedeutungslose Aktualisierung* (g_b): Ein Vertrauensaktualisierungsereignis, das keine Bedeutung im Hinblick auf das Vertrauensverhältnis hat.
- *Aktualisierung mit vermindertem Gewicht* (g^-): Aktualisierungen, die weniger Bedeutung als eine normale Aktualisierung haben.
- *Normale Aktualisierung* (g_n): Eine „normale“ Aktualisierung mit durchschnittlichen Gewicht. Mit diesen Aktualisierungen werden alle anderen Aktualisierungen gemessen, um ihnen ein vermindertes oder verstärktes Gewicht zuzuweisen.
- *Aktualisierung mit verstärktem Gewicht* (g^+): Aktualisierungen, die wichtiger als normale Aktualisierung sind.

Dabei sind g_b und g_n feste Stufen, g^- beliebige Werte zwischen g_b und g_n und g^+ beliebige Werte über g_n, wie in Abbildung 3.1 dargestellt.

3.5. Kontexte

Kontexte wurden bereits im Abschnitt 2.5 eingeführt. Im UniTEC werden Kontexte in Form von Kategorien repräsentiert.

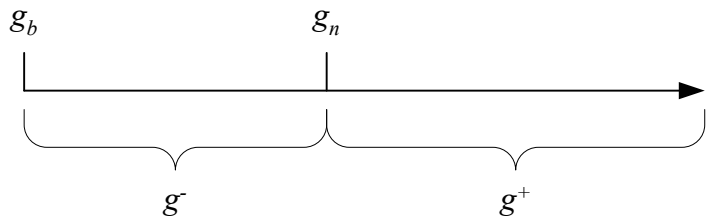


Abbildung 3.1.: Aktualisierungsgewichte

Bei Vertrauensmodellen, die unterschiedlichen Kontexte nicht vorsehen, kann der Konstrukt der Kategorie künstlich hinzugefügt werden, indem sämtliche Berechnungen nur in der Kategorie durchgeführt werden, in der die Erfahrungen gemacht wurden.

Propagierung von Bewertungen an semantisch verwandte Kategorien

Eine Vertrauensaktualisierung sollte wenn möglich an die semantisch verwandten Kategorien mit der entsprechenden Gewichtung propagiert werden, so dass eine Erfahrung gleichzeitig mehrere verwandte Kontexte beeinflussen kann. Wenn z.B. mit jemanden im Bereich „Pkw-Mechanik“ gute Erfahrungen gemacht wurden, ist zu erwarten, dass diese sich auch im allgemeineren Bereich „Mechanik“ gut auskennt. Dieses ermöglicht ein schnelleres Lernen der Vertrauensbeziehungen, indem insgesamt weniger Erfahrungen notwendig sind.

3.6. Vertrauensziel

Der Begriff des Vertrauensziels wurde bereits im Abschnitt 2.1 definiert. In einigen Forschungsarbeiten an Vertrauensmodellen zum Einsatz in Computersystemen werden unterschiedliche Vertrauensziele bereits berücksichtigt, z.B. in den Arbeiten [Jøsang u. a., 2003a], [Abdul-Rahman und Hailes, 2000] und [Sabater und Sierra, 2002]. In anderen Arbeiten wird stets davon ausgegangen, dass eine Vertrauensaussage in einem bestimmten Kontext für alle Ziele gleich ist.

Falls ein Vertrauensmodell keine Vertrauensziele vorsieht, wird diese Dimension ignoriert und es gilt die gleiche Vertrauensbeziehung für alle Ziele. Alternativ kann zu dem Vertrauensmodell die Dimension des Zieles auf ähnlicher Weise wie die Kategorien künstlich hinzugefügt und damit auch getrennt gespeichert werden.

3.7. Erfahrungen (Bewertungen von Interaktionen)

Interaktionen zwischen einem Aktor und dem Vertrauten (in UniTEC in Form von Empfehlungen) werden durch den Aktor bewertet. Eine Bewertung ordnet jeweils einem *Kriterium* einen *Bewertungswert* zu.

Bewertungskriterium

Im UniTEC-System wird eine Empfehlung im Allgemeinen mit dem einzigen Kriterium der *Nützlichkeit* bewertet. Eine Bewertung sollte also so abgegeben werden, dass diese den „Abstand“ der Meinung des Empfehlers zur eigenen Meinung zu diesem bestimmten Objekt widerspiegelt. Falls Bewertungen in dieser Weise abgegeben werden, wird sich das sich daraus errech-

nete Vertrauensverhältnis an diesen Zweck halten: Positives Vertrauen bedeutet dann, dass der Empfehler mit großer Wahrscheinlichkeit eine Empfehlung abgeben wird, die der Bewertende von Nutzen sein wird (die also der Meinung der Bewertende entspricht).

In vielen Anwendungen ist die Bewertung mit einem Kriterium ausreichend, jedoch existieren Anwendungen, wo eine Empfehlung mehrere Bewertungen zu unterschiedlichen Kriterien (z.B. Verständlichkeit, Nützlichkeit, Darstellungsform) bekommen kann. Zwei Möglichkeiten ergeben sich bei der Betrachtung solcher Bewertungen: Entweder wird ein einheitlicher Bewertungswert aus den unterschiedlichen Kriterien erstellt (z.B. durch eine gewichtete Durchschnittsbildung, die die Wichtigkeit der unterschiedlichen Kriterien berücksichtigt) oder es wird jedes Kriterium getrennt betrachtet. In diesem Fall ergeben die möglichen Kriterien eine neue Dimension in der Kontextbetrachtung. Jede Kategorie, in der die Bewertung abgegeben wird, wird zusätzlich durch das Kriterium qualifiziert.

Bewertungsmetrik

Eine Bewertungsmetrik reicht von einem Minimum (z.B. *sehr schlecht* für „Empfehlung ist überhaupt nicht nützlich“) bis zu einem Supremum (z.B. *ausgezeichnet* für „Empfehlung könnte nützlicher nicht sein“). Unterschiedliche Metriken sind dazu möglich:

- *Skalare Metrik* mit festgelegtem Minimum und Maximum. z.B. ein beliebiger Wert im Bereich $[-1..1]$.
- *Sprachliche Konstrukte*, die nach dem Bewertungsgrad geordnet werden können: z.B. *sehr schlecht*, *schlecht*, *mittelmäßig*, *gut* und *sehr gut*.
- *Diskrete Metrik*, wo eine fest definierte Anzahl von Bewertungswerten und deren Semantiken festgelegt ist. Hierzu zählen z.B. Anzahl von Sternen (1 bis 5) oder das Schulnotensystem (6 bis 1). Die einzelnen Bewertungsgrade entsprechen meistens auch sprachlichen Konstrukten.
- *Binäre Metrik* als besondere Form der Diskreten Metrik, wo nur zwei mögliche Bewertungsgrade existieren: Gut oder schlecht.

3.8. Alterung

Das Verhalten des Bewertenden kann sich im Laufe der Zeit ändern. Ältere Erfahrungen reflektieren einen vergangenen Zustand der eventuell nicht mehr mit dem aktuellen übereinstimmt (falls der Bewertende z.B. mit der Zeit bessere Empfehlungen abgibt). Daher sollte im Vertrauensmodell vorgesehen sein, dass ältere Bewertungen weniger Gewicht zugeteilt bekommen als neuere. Eine Möglichkeit dazu ist über einen *Alterungsfaktor*, der ein allmähliches „Vergessen“ älterer Erfahrungen ermöglicht (vgl. Sabater und Sierra [2000], Jøsang und Ismail [2002], Dellarocas [2003a], Huberman und Wu [2003], Kinateder und Rothermel [2003]). Eine weitere Möglichkeit ist das Zeitfenster der berücksichtigten Bewertungen durch ein *Bewertungshistorienfenster* zu beschränken (siehe „Memory Span“ in Marsh [1994], Huberman und Wu [2003]).

3.8.1. Alterungsfaktor

Ein Alterungsfaktor legt fest, wie stark neue Erfahrungen im vergleich zu älteren in einer Vertrauensberechnung berücksichtigt werden. Der Faktor beeinflusst somit direkt wie schnell auf neue Erfahrungen mit einer Vertrauensänderung reagiert wird.

Ein Alterungsfaktor beschreibt beliebige Abstufungen zwischen folgenden Wirkungen:

- *Kein Alterungseffekt*, d.h. alle Bewertungen bekommen dieselbe Gewichtung.
- *Totaler Alterungseffekt*, wo jeweils nur die neuste Bewertung berücksichtigt wird.

3.8.2. Bewertungshistorienfenster

Die Historie der Bewertungen, die im Laufe der Zeit abgegeben worden sind, kann durch ein zeitliches Fenster beschränkt werden. Ältere Bewertungen werden somit in den Berechnungen nicht mehr berücksichtigt und sind somit „vergessen“. Mögliche Varianten dieses Fenster zu beschreiben sind *reihenfolgenorientiert* (z.B. „die letzten 100 Bewertungen“), *zeitorientiert* (z.B. „alle Bewertungen der letzten 6 Monate“) oder *unbegrenzt*.

3.9. Abnutzung

Wenn in einem grösseren Zeitraum keine neuen Erkenntnisse zu einem fremden Pseudonym eintreffen, kann sich das Vertrauensverhältnis gegebenenfalls auch verändern. Dies kommt daher, dass der alte Vertrauenswert nicht unbedingt der aktuellen Realität entspricht. Die Abnutzung mit der Zeit bedeutet z.B. eine Verringerung der Vertrauensgewissheit oder sogar die Verringerung des Vertrauenswertes an sich.

Die Wirkung der Abnutzung kann durch einem *Abnutzungsfaktor* beschrieben werden. Dieser erlaubt beliebige Abstufungen zwischen folgenden Wirkungen:

- *Keine Abnutzung*, d.h. das zuletzt errechnete Vertrauensverhältnis bleibt bei Nichteintreffen anderer Vertrauensaktualisierungsereignisse erhalten.
- *Zeitliche Grenze*, d.h. es wird ein Zeitpunkt in der Zukunft definiert, wann der Vertrauenswert wieder auf eine vom Vertrauensmodell definiertes Vertrauensverhältnis zurückfällt, wenn keine weiteren vertrauensaktualisierende Ereignisse eintreffen. Im Zeitintervall zwischen der letzten Aktualisierung und der festgelegten zeitlichen Grenze fällt das Vertrauensverhältnis allmählich in diese Richtung zurück.

Da dieser Abnutzungsfaktor in dieser Arbeit neu eingeführt wird, wird dessen Wirkung im Zusammenhang mit seiner Definition für das generische Vertrauensmodell im Abschnitt 5.2.2 detaillierter beschrieben.

3.10. Vertrauensaktualisierungsalgorithmen

Um die Vertrauensaktualisierungsereignisse (in Abschnitt 3.3 eingeführt) zu verarbeiten, bieten Vertrauensmodelle aus der Literatur unterschiedliche Vertrauensaktualisierungsalgorithmen. Diese verarbeiten Ereignisse, verwenden dazu bei Bedarf auch die vergangene Bewertungshistorie oder andere Daten und verändern anhand dieser Informationen eine oder mehrere Vertrauensverhältnisse. Durch diese Veränderung entsteht die Vertrauensdynamik im Laufe der Zeit.

In der Literatur werden meistens zu jedem Vertrauensmodell ein oder mehrere Vertrauensaktualisierungsalgorithmen beschrieben.

3.11. Erwartungen

Ein Vertrauensmodell soll das menschliche Vertrauen nachbilden können, wobei es auf Erkenntnisse der Soziologie, Psychologie, Statistik oder Spieletheorie beruhen kann. Einige Eigenschaften werden von Vertrauensmodellen erwartet.

3.11.1. Umfang

Ein Vertrauensmodell sollte möglichst umfangreich sein. Die Flexibilität des Vertrauensmodells bestimmt meistens den möglichen Anwendungskreis, dem dieses Modell nutzen kann. Einfache Anwendungen kommen meistens mit wenigen Aspekten aus, jedoch erfordern moderne Anwendungen, die sich mit Vertrauensinformationen auseinandersetzen, mehr vom eingesetzten Vertrauensmodell.

3.11.2. Vertrauensinitialisierung

In anonymen Vertrauensgemeinschaften, wie es bei UniTEC im anonymen Peer-to-Peer Netz der Fall ist (siehe dazu Widmaier [2004]), kommt die Schwierigkeit hinzu, dass es eventuell möglich ist, ein verlorenes Vertrauen durch Erstellen einer neuen anonymen Identität aufzuheben. Diese Art von Angriff soll verhindert werden. Durch das Erstellen neuer Identitäten soll es nicht möglich sein, einem Vertrauensverlust entgegenzuwirken. Eine einfache Möglichkeit, dies zu erreichen ist das Vertrauensverhältnis mit dem absoluten Misstrauen zu initialisieren (wie in Kinateder und Rothermel [2003] beschrieben). Somit wird der Anreiz des Identitätswechsels entfernt. Eine weitere Möglichkeit besteht darin, das Vertrauen durch die Antworten auf eine Reputationsanfrage zu initialisieren. Falls die neue Identität in der vertrauten Gruppe unbekannt ist, wird das Vertrauensverhältnis auf das absolute Misstrauen initialisiert. Andernfalls kann aus der Reputation das Vertrauensverhältnis abgeleitet werden (z.B. durch eine nach Vertrauen gewichteten Aggregation der mitgeteilten Vertrauensinformationen). Diese Art von Lösungen werden z.B. in den Arbeiten [Yu und Singh, 2000], [Yu und Singh, 2002], [Sabater, 2003], [Abdul-Rahman und Hailes, 2000] verfolgt.

Reputationsanfrage

3.11.3. Verhalten der Vertrauensmetrik

Bei stetigem Verhalten (siehe Abschnitt 2.6) eines Empfehlers sollte das Vertrauensverhältnis nach einer gewissen Anzahl von Interaktionen relativ konstant bleiben.

Bei unehrlichen Verhalten des Empfehlers sollte der Vertrauenswert relativ schnell nach dem „Entdecken“ der Unehrlichkeit absinken, um weiteres Ausnutzen des ergatterten Vertrauens zu verhindern. Das Entdecken der Unehrlichkeit ist hier eine der Aufgaben eines Vertrauensaktualisierungsalgorithmuses. Die einfachste Form, diese Erwartungen zu erfüllen ist eine stärkere Gewichtung von negativen und schwächeren Gewichtung von positiven Erfahrungen. Damit wird eine schnelle negative und langsame positive Dynamik gewährleistet (siehe Abschnitt 2.4), welche die Erwartungen erfüllt.

3.11.4. Verhalten der Vertrauensgewissheit

Bei der Vertrauensinitialisierung sollte die Gewissheit in die Vertrauensaussage zunächst auf den minimalsten Wert der Metrik gesetzt werden, es sei denn, die Vertrauensinitialisierung beruht auf Informationen, die von Dritten stammen. In diesem Fall sollte die Gewissheit so berechnet werden, dass die Gewissheit dieser Dritten in ihren eigenen Vertrauensinformationenen aggregiert werden. Dabei kann die jeweiligen Gewissheiten zusätzlich durch die Vertrauensbeziehung zu jeden dieser Dritten gewichtet werden.

Falls die bewertende Empfehlerin ein stetiges Verhalten aufweist, sollte die Gewissheit in die Vertrauensaussage mit zunehmender Anzahl von Erfahrungen steigen. Die Anzahl von stetigen Erfahrungen, die notwendig sind, um die Gewissheit auf ein maximales Niveau zu bringen sind stark anwendungs- und benutzerabhängig.

4. Vertrauensmodelle und -algorithmen aus der Literatur

4.1. Kriterien zur Auswahl

Wir wollen uns bei der Vertrauensberechnung auf Algorithmen beschränken, die eine subjektive und lokale Sicht von Vertrauen vermitteln, so dass Algorithmen mit einer globalen oder zentralen Sicht von Vertrauen (was wir als Reputation verstehen, siehe Abschnitt 2.1.2) ausscheiden. Darunter fällt die Arbeit von [Aberer und Despotovic, 2001], wo die Vertrauensberechnung auf global verfügbare Informationen in einem verteilten Netz beruht.

Die restlichen Vertrauensmodelle werden zunächst in zwei Kategorien klassifiziert (vgl. Sabater [2003, 2.2.1]): die rein numerischen und die kognitiven Modelle. Bei den *kognitiven Modellen* werden die Faktoren beschrieben, die bei Menschen eine Vertrauensentscheidung beeinflussen. Die Vertrauensberechnung beruht auf einer Schlussfolgerung aus diesen kognitiven Eigenschaften (Macy und Skvoretz [1996], Castelfranchi und Falcone [2000]). Die rein *numerischen Modelle* konzentrieren sich auf die Art, wie sich Vertrauen letztendlich äußert: Es wird hier nicht die menschliche Denkweise nachgebildet, um daraus Vertrauensschlüsse ziehen zu können, sondern es wird versucht aus externen Eigenschaften (Erfahrungen, Bewertungen, Beobachtungen, usw.) auf Vertrauensbeziehungen zu kommen, womit eine möglichst menschliche Handelsweise nachgebildet werden kann. Diese Herangehensweise beruht auf weniger komplexen Eingangsinformationen. Bei unserer Suche nach einem generischen Vertrauensmodell betrachten wir lediglich die numerischen Vertrauensmodelle: Kognitive Modelle sind weitaus zu komplex, um sie generisch darstellen zu können.

kognitive Vertrauensmodelle

numerische Vertrauensmodelle

4.2. Vertrauensmodelle

Folgende numerische Vertrauensmodelle und die dazugehörige Vertrauensaktualisierungsalgorithmen werden in dieser Arbeit berücksichtigt und werden im Folgenden als *relevante Vertrauensmodelle und -Algorithmen* bezeichnet:

relevante Vertrauensmodelle

- Aus [Jonker und Treur, 1999] das quantitative Beispiel eines Vertrauensmodells.
- Bin Yu's erste Variante eines Vertrauensmodells aus [Yu und Singh, 2000].
- Das oft zitierte Vertrauensmodell aus [Abdul-Rahman und Hailes, 2000].

- Die Neugruppierung der Ideen von Bin Yu in [Yu und Singh, 2002].
- Das *Beta Reputation System* aus [Jøsang und Ismail, 2002].
- Das *ReGreT System* aus [Sabater, 2003].
- Die Arbeit zu Vertrauen in [Mui, 2003].
- Die Vertrauensberechnung laut [Kinateder und Rothermel, 2003].

Das Vertrauensmodell von [Marsh, 1994] wird in einigen Arbeiten als Grundlage zitiert und dient deswegen auch als Basis zur Erarbeitung des generischen Vertrauensmodells.

nicht lokale Informationen

Bei einigen Vertrauensmodellen sind wichtige Bestandteile der Vertrauensberechnung neben den direkten Erfahrungen auch fremde Aussagen (Yu und Singh [2000], Abdul-Rahman und Hailes [2000], Yu und Singh [2002], Sabater [2003]) oder Analysen des sozialen Netzes in dem ein Agent eingebunden ist (Sabater [2003], Mui [2003]). Diese Möglichkeiten sollen den Algorithmen ebenfalls im generischen Vertrauensmodell und im UniTEC-System zur Verfügung stehen.

Im Folgenden werden Details zu den relevanten Vertrauensmodellen und speziell zu deren lokalen Vertrauensaktualisierungsalgorithmen vorgestellt. Wir legen zunächst folgende einheitliche Bezeichnungen fest:

b_i Wert der Bewertung i.

T_i Vertrauenswert nach Berücksichtigung aller Bewertungen b_j mit $0 < j \leq i$.

G_i Gewissheit des Vertrauenswertes nach Berücksichtigung aller Bewertungen b_j mit $0 < j \leq i$.

Alle Werte verwenden jeweils die Metrik des entsprechenden Vertrauensmodells. Diese Bezeichnungen werden ab hier durchgängig verwendet, wobei bei allen Erläuterungen jeweils ein Vertrauensverhältnis betrachtet wird (d.h. zwischen genau zwei Entitäten in einem Kontext und zu einem Vertrauensziel). Die Originalbezeichnungen aus der Literatur werden entsprechend auf diese Bezeichnungen angepasst.

Nicht jedes in der Literatur beschriebene Vertrauensmodell befasst sich mit allen Merkmalen, die auch in Kapitel 3 beschrieben wurden.

4.2.1. Jonker [1999]

In [Jonker und Treur, 1999] werden zunächst grundsätzliche Eigenschaften von Vertrauensaktualisierungsalgorithmen besprochen. Es werden wichtige Erkenntnisse zu der Klassifikation von Vertrauensalgorithmen gemacht. Um diese Eigenschaften konkret zu demonstrieren, wird ein einfaches Vertrauensmodell und ein dazugehöriges Vertrauensaktualisierungsalgorithmus beispielhaft vorgestellt.

Metriken

In diesem Modell werden Vertrauenswerte und Erfahrungen im Bereich $[-1,+1]$ dargestellt. Bei dem Vertrauensaktualisierungsalgorithmus handelt es sich um eine Aktualisierungsfunktion, die jeweils nur auf den letzten Vertrauenswert (T_{i-1}) und der neuen Erfahrung in Form einer Bewertung (b_i) beruht und daraus den neuen Vertrauenswert (T_i) berechnet. Eine Art Alterungsfaktor wird hier „*inflation rate*“ (d) genannt, der neuen Erkenntnissen jeweils grössere Gewichtung gibt:

$$T_i = d \cdot T_{i-1} + (1-d) \cdot b_i$$

Es handelt sich hier um ein einfaches geometrisches Lernen.

4.2.2. Bin Yu [2000]

Als Grundlage der Arbeit in [Yu und Singh, 2000] werden die Definitionen von [Marsh, 1994] zitiert. Die drei wichtigen Komponenten zur Vertrauensbildung sind eigene Erfahrungen, Zeugenaussagen (*witness testimony*) und Gerüchteverbreitung (*gossip*).

Der lokale Vertrauensaktualisierungsalgorithmus basiert – wie auch Jonker [1999] – auf den zuletzt gespeicherten Vertrauenswert und einer neuen Bewertung. Hier ist die Rechenvorschrift allerdings etwas komplexer.

Bewertungsmetrik

Jede Bewertung wird zunächst als positive Erfahrung (*cooperation*) oder negative Erfahrung (*defection*) eingestuft. Im Fall einer negativen Erfahrung, wird die Bewertung in Form eines Wertes $\beta < 0$ gekennzeichnet, im Fall einer positiven Erfahrung, $\alpha > 0$. Die Skalierung der positiven und negativen Bewertung kann unterschiedlich sein. Diese unterschiedliche Skalierung beeinflusst direkt die Funktionsweise des Algorithmus. Im Paper wird z.B. empfohlen, dass $|\alpha| < |\beta|$ gewählt wird, womit negative Erfahrungen stärker als positive berücksichtigt werden.

Die Aktualisierung des Vertrauens basiert dann auf die Rechenvorschrift laut Tabelle 4.1.

T_{i-1}	Positive Erfahrung α	Negative Erfahrung β
> 0	$T_{i-1} + \alpha_i(1 - T_{i-1})$	$(T_{i-1} + \beta_i)/(1 - \min\{\lvert T_{i-1}\rvert, \lvert\beta_i\rvert\})$
< 0	$(T_{i-1} + \alpha_i)/(1 - \min\{\lvert T_{i-1}\rvert, \lvert\alpha_i\rvert\})$	$T_{i-1} + \beta_i(1 + T_{i-1})$
$= 0$	α_i	β_i

Tabelle 4.1.: Berechnungsvorschrift für T_i laut [Yu und Singh, 2000]

Bei den Zeugenaussagen und Gerüchte werden die lokalen Vertrauensbeziehungen zur Vorselektion und Gewichtung der erhaltenen Informationen verwendet. Zeugenaussagen können zusätzlich über Verweisketten indirekt ankommen, wobei dann die gesamte Vertrauenskette analysiert wird.

4.2.3. Rahman [2000]

Das Vertrauensmodell in [Abdul-Rahman und Hailes, 2000] basiert auf Erkenntnissen aus der Soziologie, wie auch bei [Marsh, 1994]. Drei Vertrauensarten werden festgelegt (siehe dazu auch Abschnitt 2.1.1 und [McKnight und Chervany, 1996]): *interpersonal*, *system* und *dispositional trust*. Das Vertrauensmodell arbeitet lediglich mit Interpersonellem Vertrauen. Zusätzlich werden Kontexte definiert, in denen die Vertrauensbeziehungen gelten.

Vertrauensmetrik

Es werden vier Grade an Vertrauen festgelegt: *very untrustworthy = vu*, *untrustworthy = u*, *trustworthy = t* und *very trustworthy = vt*. Zusätzlich sind drei Unsicherheitsgrade bei direkter Erfahrung definiert: *mostly good and some bad*, *mostly bad and some good* und *equal amount of good and bad*. Das Vertrauen baut sich aus direktem Vertrauen (*direct trust*) und Empfehlervertrauen (*recommender trust*) auf.

Bewertungsmetrik

Erfahrungen beeinflussen das direkte Vertrauen und werden in Form von Bewertungen festgehalten, die ebenfalls einer diskreten Metrik mit vier möglichen Bewertungswerte unterliegen: *very bad = vb*, *bad = b*, *good = g* und *very good = vg*.

Zur Bestimmung des direkten Vertrauens wird zu jedem Wert der Bewertungsmetrik ein Zähler geführt, der angibt, wie oft eine Bewertung mit dem jeweiligen Wert abgegeben wurde. Um das direkte Vertrauen (*direct trust*) zu ermitteln, wird der momentan höchste Zähler betrachtet, und dessen Position als Vertrauensgrad verwendet (*vb = very untrustworthy*, *bad = untrustworthy*, *good = trustworthy* und *very good = very trustworthy*). Falls mehrere höchste Zähler existieren, wird ein Unsicherheitswert ausgegeben, laut Vorschriften der Tabelle 4.2.

Höchste Zähler	Unsicherheitswert	Bedeutung
vg, *g* und ein weiterer	u^+	Meist gute, einige schlechte Erfahrungen
vb, *b* und ein weiterer	u^-	Meist schlechte, einige gute Erfahrungen
Alle anderen Kombinationen	u^0	Gleichermaßen gute und schlechte Erfahrungen

Tabelle 4.2.: Unsicherheitswert laut [Abdul-Rahman und Hailes, 2000]

Zur Bestimmung des Empfehlungsvertrauens wird die semantische Distanz zu jedem Vertrauensempfehler betrachtet. Diese beschreibt, wie oft in der Vergangenheit Empfehlungen des Vertrauensempfehlers zu guten Bewertungen führten. Durch diese semantische Distanz wird nun die eigentliche Vertrauensempfehlung evaluiert und als neue Erfahrung auf lokale Vertrauensberechnungen hinzugefügt.

4.2.4. Bin Yu [2002]

Das Vertrauensmodell von [Yu und Singh, 2002] basiert auf der Dempster-Shafer-Theorie. In dieser Theorie werden positive und negative Indizien berücksichtigt. Es gibt keine direkte Be-

ziehung zwischen einer Hypothese und ihrer Negation. Somit kann mangelnde Überzeugung (*belief*) nicht direkt in Unglaube (*disbelief*) übertragen werden. Mangelnde Überzeugung und mangelnder Unglaube werden als Ungewissheit dargestellt.

In der Umsetzung des Dempster-Shafer Ansatzes betrachten Bin Yu die Glaubwürdigkeit der Hypothese „positives Vertrauen“ (V). Als Negation dazu gilt $\neg V$. Die betrachtete Beurteilungsmenge (*frame of discernment*) ist somit $\Theta = \{V, \neg V\}$.

Für die lokale Vertrauensberechnung (*local trust ratings*) wird jede Begegnung mit einen fremden Agenten mit einer Bewertung versehen (Bin Yu nennt es *QoS = Quality of Service*). Eine Bewertung kann eine von elf diskreten Werten $\{0.0, 0.1, ...1.0\}$ einnehmen. Nach [Marsh, 1994] wird auch hier ein unterer Schwellwert (ω) und ein Oberschwellwert (Ω) für das Vertrauen definiert (mit $0 \leq \omega \leq \Omega \leq 1$). Unter dem unteren Schwellwert wird nicht vertraut, über den Oberschwellwert wird vertraut, und dazwischen ist der Zustand undefiniert. Bewertungsmetrik

Die relativen Häufigkeiten der einzelnen Bewertungen werden nun berechnet:

$$f(x_k) = \frac{h_{x_k}}{H}$$

wobei $x_k \in \{0.0, 0.1, ..., 1.0\}$ die einzelne Bewertungsgrade sind, h_{x_k} die Anzahl der Bewertungen mit dem Wert x_k und H die Gesamtanzahl der Bewertungen ist.

Folgende Wahrscheinlichkeitszuordnungen (*basic probability assignment = bpa*) werden festgelegt:

$$m(\{\neg V\}) = \sum_{x_k=0}^{\omega} f(x_k)$$
$$m(\{V, \neg V\}) = \sum_{x_k=\omega}^{\Omega} f(x_k)$$
$$m(\{V\}) = \sum_{x_k=\Omega}^{1} f(x_k)$$

Somit ist die Glaubwürdigkeitsfunktion (*belief function*) $Bel(\{V\}) = m(\{V\}) = T$ die Angabe des Grades der Überzeugung, dass dem fremden Agent vertraut werden kann ([0..1]).

Zudem wird die Dempster’s Zusammenführungsregel (*Rule of Combination*) verwendet, um Erfahrung unterschiedlicher Agenten miteinander zu kombinieren.

4.2.5. Jøsang [2002] (Beta Reputation System)

Das *Beta Reputation System* aus [Jøsang und Ismail, 2002] basiert auf der Bayeschen Wahrscheinlichkeit. Es verwendet die Beta-Verteilung, die die Wahrscheinlichkeitsverteilung binärer

Ereignisse (hier die eingetroffenen Bewertungen) darstellen kann. Nach jeder eingetroffene Bewertung kann die *posteriori* Wahrscheinlichkeit als eine Beta-Verteilung dargestellt werden. Diese mathematisch fundierte Basis wird verwendet, um Bewertungen in Vertrauensaussagen zusammenzufassen.

Die entsprechende Beta-Verteilung wird durch eine Funktion der Beta-Familie beschrieben:

$$f(p|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \cdot p^{\alpha-1} \cdot (1-p)^{\beta-1}$$

Eine detaillierte Herleitung der Beta Funktion ist in [Weisstein, 2004a] zu finden. Die Beschreibung und Herleitung der Gamma Funktion wird in [Weisstein, 2004b] beschrieben. Ausführliche Informationen zu der Beta-Verteilung und Ihren Einsatz in der Entscheidungstheorie werden in [Winkler, 1972] gezeigt.

In r werden die positiven und in s die negativen Erfahrungen festgehalten. Diese bilden sich aus einer Akkumulierung (entsprechend gewichtet und gealtert) der einzelnen Bewertungen (r_i und s_i). Die Beta-Verteilung, die diese Erfahrungen widerspiegelt ergibt sich durch folgende Substitution:

$$\alpha = r+1 \qquad \text{und} \qquad \beta = s+1$$

Der Erwartungswert der Wahrscheinlichkeit p in dieser Beta-Verteilung wird nun als Vertrauenswert definiert und wird wie folgt berechnet:

$$T_i = \frac{r(i)+1}{r(i)+s(i)+2}$$

Zusätzlich wird in [Jøsang und McAnally, 2004] die Ungewissheit der Vertrauensaussage als Varianz der Beta-Verteilung definiert.

4.2.6. Sabater [2003] (ReGreT)

Im ReGreT System aus [Sabater, 2003] wird ein Reputationssystem beschrieben, welches Informationen aus lokalen Erfahrungen, Zeugenaussagen und aus dem sozialen Netz, in dem die Beteiligten miteinander kommunizieren, benutzt. Eine ontologische Dimension beschreibt unterschiedliche Vertrauensziele.

Bewertungsmetrik

Die lokale Vertrauensberechnung (*direct trust*) wird aus einer Aggregation der Bewertungen aus einer Erfahrungsdatenbank erzeugt. Einzelne Bewertungen sind auf dem Wertebereich $[-1,+1]$ eingeschränkt. Zur Berechnung werden alle relevanten Bewertungen aus der Datenbank in Form eines gewichteten Durchschnitts aggregiert:

$$T_i = \sum_{j=0}^{i} \rho(t,t_j) \cdot b_j$$

Die Funktion $\rho(t,t_j)$ realisiert eine gewichtete Durchschnittsbildung. Dabei sollen neuere Erfahrungen stärker gewichtet werden als ältere. Es muss gewährleistet sein, dass $\sum_{j=0}^{t} \rho(t,t_j) = 1$.

In [Sabater, 2003] wird ein Beispiel einer solchen Alterungsfunktion angegeben:

$$\rho(t,t_j) = \frac{f(t_j,t)}{\sum_{k=0}^{i} f(t_k,t)} \qquad \text{mit} \qquad f(t_j,t) = \frac{t_j}{t}$$

Hier wird eine proportionale Alterung definiert, wobei die erste Bewertung die Gewichtung 0 und die neuste Bewertung die Gewichtung 1 bekommt.

Es wird zu jeder Vertrauensberechnung zusätzlich eine Gewissheit berechnet. Diese bildet sich aus zwei Faktoren: aus der Anzahl an Erfahrungen (*number of outcomes factor* No_i) und aus Abweichungen der Bewertungen zu dem Vertrauen (*outcome reputation deviation* Dv_i).

$$\begin{aligned} No_i &= \begin{cases} \sin(\frac{\pi}{2}) \cdot \frac{i}{itm} & \text{wenn} \quad i \leq itm \\ 0 & \text{sonst} \end{cases} \\ Dv_i &= 1 - \sum_{j=0}^{i} \rho(t,t_j) \cdot |b_j - T_i| \end{aligned}$$

Dabei stellt itm (*intimate level of interactions*) die Anzahl von Erfahrungen dar, die notwendig sind, damit ein Empfehler als Bekannt gilt. Bis dorthin wächst der Faktor No_i Sinusförmig zu 1.

Gewissheitsmetrik

Beide Faktoren werden in einem Gewissheitswert (*reliability*) $[0..1]$) für das direkte Vertrauen zusammengefasst:

$$G_i = No_i \cdot (1 - Dv_i)$$

Zusätzlich zu dieser lokalen Vertrauensberechnung wird ein Reputationswert aus Zeugenaussagen gebildet. Ein Kredibilitätsmodell beschreibt inwieweit welchen Agenten geglaubt werden kann. Es wird beschrieben, wie diese Informationen zusammengefügt werden können.

Die ontologische Dimension bildet die letzte Komponente der Reputation. Hier werden die einzelnen Vertrauensziele beschrieben, die für eine Transaktion relevant sind (z.B. „kann schnell liefern“, „bietet gute Preise“, „bietet gute Qualität“). Das allgemeine Vertrauen wird aus einer Kombination der einzelnen Vertrauensziele gebildet.

4.2.7. Lik Mui [2003]

Das Vertrauensmodell und die lokale Vertrauensberechnung in [Mui, 2003] basiert auf ähnliche Grundlagen wie [Jøsang und Ismail, 2002]. Auch hier wird eine Wahrscheinlichkeitsverteilung in Form einer Beta-Verteilung aufgestellt. Aus dieser Verteilung kann ein Erwartungswert berechnet werden, der als Vertrauenswert für die nächste Begegnung verwendet wird.

Anders als bei [Jøsang und Ismail, 2002] werden jedoch hier nur binäre Bewertungen zugelassen und eine Alterung oder Gewichtung der Bewertungen ist nicht vorgesehen. Diese Einschränkungen machen aus diesem lokalen Teil der Vertrauensaktualisierungsalgorithmus einen Spezialfall der Vertrauensberechnung aus [Jøsang und Ismail, 2002]. Daher wurde für unsere Analysen dieser Algorithmus nicht getrennt implementiert.

4.2.8. UniTEC [2003]

Vertrauensmetrik

Bewertungsmetrik

Im ersten Paper zu UniTEC (Kinateder und Rothermel [2003]) wird ein Reputationssystem vorgestellt, welches zur Vertrauensberechnung auf einem Vertrauensmodell beruht, das Vertrauenswerte im Bereich $[0..1]$ in unterschiedlichen Kontexten vorsieht. Der Vertrauensaktualisierungsalgorithmus berechnet aufgrund des zuletzt gespeicherten Vertrauenswertes T_i und einer neuer Bewertung b (die auf die binäre Werte 0 und 1 eingeschränkt ist) ein neuen Vertrauenswert T_{i+1} nach folgender Vorschrift:

$$T_i = (1 - (\alpha \cdot g_i)) \cdot T_{i-1} + (\alpha \cdot g_i) \cdot b$$

Dabei ist α der Alterungsfaktor im Bereich $]0,1[$ und g_i das Gewicht der Aktualisierung i ($[0..1]$).

Auch hier beschreibt die Berechnung wie bei [Jonker und Treur, 1999] ein geometrisches Lernen, wobei hier die Gewichtung der Bewertungen explizit eine Rolle spielt (diese bildet sich aus semantischer Distanz der Kategorie und der Empfehlergewissheit).

Die Gewissheit der Vertrauensaussage wird in Form eines Vektors mit der Anzahl direkter und indirekter Erfahrungen beschrieben, es wird keine Metrik beschrieben, wie diese Daten zu interpretieren sind.

4.3. Zusammenfassung

Die Grundeigenschaften der besprochenen Vertrauensmodelle sind in Tabelle 4.3 zusammengefasst. Des Weiteren wird eine Übersicht der unterschiedlichen Bewertungs- (Tabelle 4.4), Vertrauens- und Gewissheitsmetriken (Tabelle 4.5) aufgestellt.

Modell	**Informationsquelle** [b]	**Vertrauensaktion**	**Mathematisches Modell**
Marsh [1994]	Direkte Erfahrungen	Kooperiere	Wahrscheinlichkeit
Castelfranchi	Nicht angegeben	Delegiere	Skalare Aggregation
Jonker [1999]	Direkte Erfahrungen	Finde Vertrauenswürdige[a]	Skalare Aggregation (geometrisches Lernen)
Bin Yu [2000]	Direkte Erfahrungen, Fremde Aussagen	Kooperiere	Skalare Aggregation
Rahman [2000]	Direkte Erfahrungen, Fremde Aussagen	Finde Vertrauenswürdige	Diskrete Zählung
Bin Yu [2002]	Direkte Erfahrungen, Fremde Aussagen	Finde Vertrauenswürdige	Dempster-Shafer
Beta Reputation System	Direkte Erfahrungen	Finde Vertrauenswürdige[a]	Dempster-Shafer und Bayesche Wahrscheinlichkeit
ReGreT	Direkte Erfahrungen, Fremde Aussagen, Soziales Netz, Vorurteile	Finde Vertrauenswürdige[a]	Skalare Aggregation (Durchschnittsbildung)
Lik Mui [2003]	Direkte Erfahrungen, Soziales Netz	Kooperiere	Bayesche Wahrscheinlichkeit
UniTEC [2003]	Direkte Erfahrungen	Finde Vertrauenswürdige	Skalare Aggregation (geometrisches Lernen)

Tabelle 4.3.: Eigenschaften der Vertrauensmodelle aus der Literatur

[a] In der Literatur wird diese Eigenschaft nicht explizit erwähnt.
[b] Vgl. Sabater [2003]

Modell	**Bewertungsmetrik**
Marsh [1994]	Binär *Cooperate/Defect*
Jonker [1999]	Skalar: $[-1..+1]$ *Experience value*
Bin Yu [2000]	Gemischt Binär: *Cooperation/Defection* Skalar: *Positive* $\alpha \geq 0$ und *negative* $\beta \leq 0$ *evidence* (S.5)
Rahman [2000]	Diskret: 4 Werte vg (*very good*), g, b, vb (*very bad*) (Abdul-Rahman und Hailes [2000, S.5])
Bin Yu [2002]	Diskret: 11 Werte: $0.0, 0.1, ...1.0$ (2.2 S.3)
Beta Reputation System	Skalar *Positive Feedback* $r > 0$, *Negative Feedback* $s > 0$
ReGreT	Skalar: $[-1..+1]$
Lik Mui [2003]	Binär (*positive/negative encounter*) (Mui u. a. [2001, S.3])
UniTEC [2003]	Binär (*good/bad*)

Tabelle 4.4.: Bewertungsmetriken der Vertrauensmodelle aus der Literatur

Modell	**Vertrauensmetrik**	**Gewissheitsmetrik**
Marsh [1994]	Skalar: $[-1..+1)$	Keine
Jonker [1999]	Skalar: $[-1..+1]$ *Trust value*	Keine
Bin Yu [2000]	Skalar: $[-1..+1)$ *Trust rating* (S.5 Def 2)	Keine
Rahman [2000]	Diskret: 4 Werte vt, t, u, vu (4.2 S.4)	Diskret: 3 Werte u^+, u^0, u^- (4.7 S.5)
Bin Yu [2002]	Skalar: $[0..1]$ *Belief Probability Assignment*	Skalar: $[0..1]$
Beta Reputation System	Skalar: $[0..1]$ *Probability expectation* *Reputation*	Skalar: $[0..1]$ Standardabweichung der Beta-Verteilung
ReGreT	Skalar: $[-1..+1]$ *Reputation*	Skalar: $[0..1]$ *Reliability* (4.4 S.46-47)
Lik Mui [2003]	Skalar: $[0..1]$ *Probability expectation*	Skalar: $[0..1]$
UniTEC [2003]	Skalar: $[0..1]$	Zähler direkter, indirekter Erfahrungen (*Confidence vector*)

Tabelle 4.5.: Vertrauensmetriken der Vertrauensmodelle aus der Literatur

5. Das generische Vertrauensmodell

Um möglichst viele Vertrauensmodelle im UniTEC-System einsetzen zu können, wurde ein generisches Modell erstellt, worauf diese Modelle abgebildet werden können. Das generische Vertrauensmodell ergibt sich aus einer Obermenge aller Eigenschaften, die von den einzusetzenden Modellen unterstützt werden. Eine intelligente Abbildung zwischen einzelnen Eigenschaften muss dazu möglich sein, um eine einheitliche Darstellung zu erlauben und Vergleiche ziehen zu können.

Abbildungen

Da es sich um ein verteiltes Agentensystem handelt, bekommt das generische Vertrauensmodell zugleich auch die Aufgabe einer globalen einheitlichen Darstellung von Vertrauensinformationen. Somit kann jeder Agent im UniTEC-System ein anderes lokales Vertrauensmodell einsetzen, und trotzdem mit anderen Agenten auf unmissverständliche Art kommunizieren.

global einheitliche Metrik

Zunächst werden die erforderlichen Metriken vereinheitlicht. Anschließend werden diverse Faktoren, die schon in Abschnitt 3 beschrieben wurden, in dieses generische Modell integriert. Erst später bei der Umsetzung der einzelnen relevanten Vertrauensmodellen (Abschnitt 7) werden einige Entscheidungen klar, die zu dem generischen Vertrauensmodell geführt haben.

5.1. Metriken

5.1.1. Vertrauensmetrik

Unter den relevanten Vertrauensmodellen finden wir folgende Varianten an Vertrauensmetriken:

- Skalare Werte im Bereich $[-1,+1]$ (Jonker und Treur [1999], Yu und Singh [2000], Sabater [2003]).
- Skalare Werte als Wahrscheinlichkeiten im Bereich $[0,1]$ (Jøsang und Ismail [2002], Yu und Singh [2002], Mui [2003], Kinateder und Rothermel [2003]).
- Diskrete Werte (Abdul-Rahman und Hailes [2000] mit vier möglichen Vertrauenswerten).

Definition 4 *In unserem generischen Vertrauensmodell definieren wir eine Vertrauensbeziehung mit einer Metrik skalarer Werte im Bereich* $[0,1]$*, wobei* 0 *das* absolute Misstrauen *und* 1 *das* blinde Vertrauen *bedeuten.*

Diese Metrik definiert eine vollständige Halbordnung.

5.1.2. Gewissheitsmetrik

Die Gewissheit spielt bei Vertrauensaussagen (in Form von Vertrauensgewissheit) und bei Empfehlungen (in Form von Empfehlungsgewissheit) eine Rolle.

In den Modellen, wo die Gewissheit bereits berücksichtigt wird, wird diese entweder in dem Bereich ($[0,1]$) repräsentiert (Yu und Singh [2002], Jøsang und Ismail [2002], Sabater [2003], Mui [2003]) oder die Gewissheit wird nicht durch einen einzelnen Wert ausgedrückt, sondern bildet sich aus mehreren Eigenschaften (Kinateder und Rothermel [2003]).

Definition 5 *Die Gewissheit im generischen Vertrauensmodell wird als skalarer Wert im Bereich* $[0,1]$ *dargestellt, wobei* 0 *die* komplette Ungewissheit *und* 1 *die* totale Sicherheit *bedeuten.*

Diese Metrik definiert ebenfalls eine vollständige Halbordnung.

5.1.3. Bewertungsmetrik

Wie auch bei der Vertrauensmetrik erscheinen hier in den relevanten Vertrauensmodellen unterschiedliche Metriken zur Bewertung einzelner Ereignisse (Begegnungen, Empfehlungen, usw.) durch den Benutzer:

- Skalare Werte im Bereich $[-1,+1]$ werden in [Jonker und Treur, 1999], [Sabater, 2003] verwendet.
- Diskrete Werte: In [Mui, 2003] und [Kinateder und Rothermel, 2003] werden die Bewertungen binär abgegeben (*gut* oder *schlecht*). Das Vertrauensmodell aus [Abdul-Rahman und Hailes, 2000] arbeitet mit vier (*very bad*, *bad*, *good* und *very good*), in [Yu und Singh, 2002] wird mit elf diskreten Werten $(0.0, 0.1, ..., 1.0)$ gearbeitet.
- Andere Varianten: In [Yu und Singh, 2000] wird jede Erfahrung zunächst als Positiv oder als Negativ eingestuft. Positive Erfahrungen (*cooperation*) bekommen einen Wert $\alpha > 0$, negative (*defection*) einen Wert $\beta < 0$. Ähnlich ist es in [Jøsang und Ismail, 2002], wo jede Bewertung aus den Faktoren r (positive Erfahrung) und s (negative Erfahrung) besteht.

Definition 6 *Die Bewertungsmetrik im generischen Vertrauensmodell wird ebenfalls als skalare Werte im Bereich* $[0,1]$ *festgelegt, wobei* 0 *eine* absolut schlechte *und* 1 *eine ausgezeichnete Bewertung bedeuten.*

Diese Metrik definiert eine vollständige Halbordnung.

5.1.4. Wertmetrik

Jede Begegnung zwischen zwei Beteiligten (im UniTEC in Form einer Anfrage und darauf geantwortete Empfehlung) bekommt einen Wert zugeordnet, der beschreibt, wie wichtig diese Begegnung (als Kooperation) für den Benutzer ist. Der Wert setzt sich aus der *Wichtigkeit* und dem *Nutzen* zusammen (siehe dazu Marsh [1994, 4.3.4]) und wird in den Bereich $[0,1]$ normalisiert. D.h. der Wert 0 symbolisiert *kein Nutzen* oder *keine Wichtigkeit*, während der Wert 1 den *maximalen Nutzen* und die *maximale Wichtigkeit* beschreibt.

Der Wert einer Begegnung wird in [Marsh, 1994] beschrieben, wird jedoch in keinem der relevanten Modelle als Eingangsparameter explizit erwähnt. Im generischen Modell wird dieser Wert zur Gewichtung der Bewertungen verwendet, so dass Modelle, die eine solche Gewichtung vorsehen, den Wert implizit verwenden (siehe auch Abschnitt 3.4).

5.2. Faktoren

5.2.1. Alterungsfaktor

In den Modellen, wo ein Alterungsfaktor verwendet wird, ist dieser stets als skalarer Wert im Bereich $[0,1]$ dargestellt.

Definition 7 *Der Alterungsfaktor im generischen Vertrauensmodell wird als ein skalarer Wert* α *mit* $0 \leq \alpha \leq 1$ *dargestellt, wobei* 1 *den* totalen Alterungseffekt *beschreibt. Ein Alterungsfaktor von* 0 *schaltet den Alterungseffekt komplett aus.*

Die Änderung dieses Faktors beeinflusst auf unterschiedliche Weise die Wirkung des entsprechenden Aktualisierungsalgorithmus. Somit stellt im Sinne der Vergleiche der Algorithmen jede Einstellung dieses Faktors bei einem bestimmten Aktualisierungsalgorithmus eine andere Variante dieses Algorithmus dar. Die Auswirkung eines bestimmten Alterungsfaktors kann bei unterschiedlichen Algorithmen verschieden sein.

Alterungsfaktor in der Literatur

Bei [Jonker und Treur, 1999] und bei [Kinateder und Rothermel, 2003] ist der Alterungsfaktor (bei Jonker und Treur als Bestandteil der „*inflation rate*“) der ausschlaggebende Faktor für den Aktualisierungsalgorithmus. In [Jøsang und Ismail, 2002] ist dieser Faktor nur eine Möglichkeit, die Funktionsweise der Vertrauensaktualisierungen zu beeinflussen, der auch komplett ausgeschaltet werden kann (bei $\alpha = 0$). In [Sabater, 2003] wird nicht ein Alterungsfaktor sondern eine komplette Alterungsfunktion als Parameter zum Algorithmus festgelegt, die beschreibt, wie die Bewertungen in der Bewertungshistorie gewichtet werden sollen. Der Aktualisierungsfaktor kann hier bei einer gegebenen Alterungsfunktion zur Abschwächung dessen Wirkung verwendet werden.

5.2.2. Abnutzungsfaktor

Keiner der relevanten Vertrauensmodelle arbeitet mit einem Abnutzungsfaktor (siehe Abschnitt 3.9). Somit wird dieser hier neu eingeführt.

Definition 8 *Der Abnutzungsfaktor ist ein skalarer Wert $\lambda \geq 0$, wobei $\lambda = 0$* kein Abnutzungseffekt *beschreibt. Je grösser der Abnutzungsfaktor, desto schneller wird das Vertrauen ohne neue Erkenntnisse „vergessen“. Die mathematische Beschreibung dieses Vergessens wird durch eine Abnutzungsfunktion beschrieben.*

Wird der Vertrauenswert zu einem beliebigem Zeitpunkt abgefragt (auch z.B. vom Algorithmus selbst im Zuge einer Vertrauensaktualisierung), kann die Abnutzungsfunktion den aktuellen Vertrauenswert berechnen:

$$T_{t+n} = f(T_t, n, \lambda)$$

Zu dem Zeitpunkt t wurde die letzte Erfahrung gemacht. Zu diesem Zeitpunkt wurde das Vertrauen berechnet: T_t, welches den Vertrauenswert und die Vertrauensgewissheit beschreibt. Nun wird das Vertrauensverhältnis zu dem Zeitpunkt $t+n$ abgefragt (T_{t+n}). Die Abnutzungsfunktion f berechnet das zu diesem Zeitpunkt gültige Vertrauensverhältnis. Bei der Berechnung fließt auch der Abnutzungsfaktor λ mit ein.

Basisvertrauensverhältnis

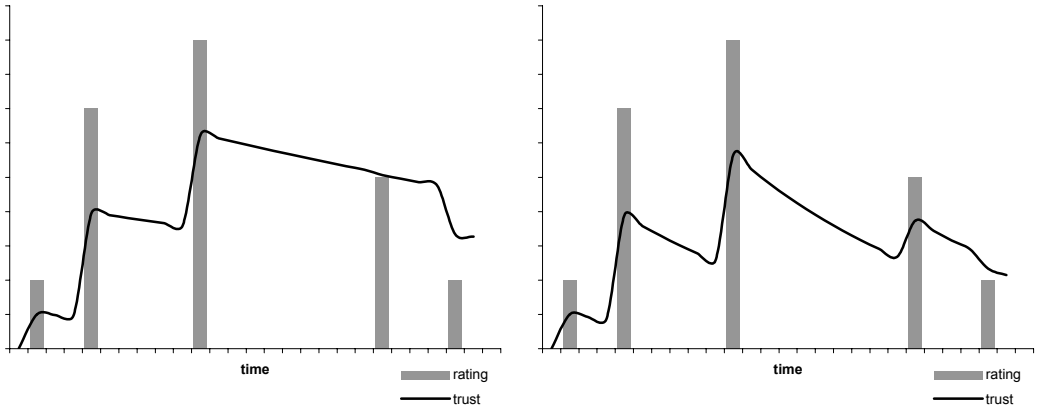
Einige Vertrauensmodelle legen ein sogenanntes *Basisvertrauensverhältnis* fest. Dieses Vertrauensverhältnis herrscht dann, wenn keine Informationen zu dem bewertenden Fremden existieren (weder in Form eigener Erfahrungen und Beobachtungen, noch in Form von Aussagen Dritter). Das Vertrauensmodell kann dazu z.B. das absolute Misstrauen, ein Vertrauensverhältnis mit einer kompletten Ungewissheit oder ein anderes vorbestimmtes Vertrauensverhältnis auswählen. Der Abnutzungsalgorithmus kann nun die Funktion f so gestalten, dass mit fortschreitender Zeit das Vertrauensverhältnis zurück auf dieses Basisvertrauensverhältnis zurückfällt.

Abnutzung und die Gewissheit

Die Abnutzungsfunktion sollte die Abnutzung mit der Vertrauensgewissheit gewichten. Ein Vertrauensverhältnis mit einer starken Gewissheit soll bei mangelnden Erfahrungen schwächer abgenutzt werden als Vertrauensverhältnisse mit schwacher Gewissheit. Eventuell kann das Vertrauensmodell einen Gewissheitsschwellwert festlegen, über dem gar keine Abnutzung mehr stattfindet. Auch zu überlegen wäre den Abnutzungsfaktor bei Neueintritt in einer Gemeinschaft für eine gewisse Startzeit komplett auszuschalten, während die Gewissheit aufgebaut wird.

Die Wirkungsweise des Abnutzungsfaktors wird nun beispielhaft gezeigt. In den Abbildungen 5.1(a) und 5.1(b) wird gezeigt, wie unterschiedliche Abnutzungsfaktoren ($\lambda = 0.1$ und $\lambda = 0.4$) die Vertrauensdynamik beeinflussen können. In beiden Fällen wird als Basisvertrauenswert das absolute Misstrauen verwendet (0). Die Gewissheit spielt in diesem Beispiel keine Rolle.

Deutlich zu erkennen ist die Abnutzung des Vertrauens nach jeder Bewertung, in der Zeit wo keine Bewertungen eingehen. In Abbildung 5.1(a) mit dem niedrigerem Faktor ist diese Abnutzung sichtlich geringer. Lediglich an den Zeitpunkten der Bewertungen wird der Verlauf der



(a) Abnutzungsfaktor $\lambda = 0.1$ (b) Abnutzungsfaktor $\lambda = 0.4$

Abbildung 5.1.: Vertrauensverlauf mit Abnutzungsfaktor (Alterungsfaktor $\alpha = 0.5$)

Kurve deutlich geändert. Schon in Abbildung 5.1(b) mit dem viel stärkeren Abnutzungsfaktor schafft es jede positive Bewertung den Vertrauenswert nur kurzzeitig „hochzuziehen“. Die Abnutzung ist so stark, dass der Vertrauenswert sehr rasch wieder zurückfällt. Die Festlegung eines Abnutzungsfaktors hängt also direkt mit der Dichte der Bewertungen zusammen, die in der bestimmten Vertrauensapplikation gemacht werden. Er muss so konzipiert sein dass er einen positiven Trend nicht zu stark negativ prägt, aber dennoch eine zu große Lücke zwischen Bewertungen genügend bestraft. Die Abnutzungsfunktion sollte den Abnutzungsfaktor als Indikator für diese Eigenschaften verwenden.

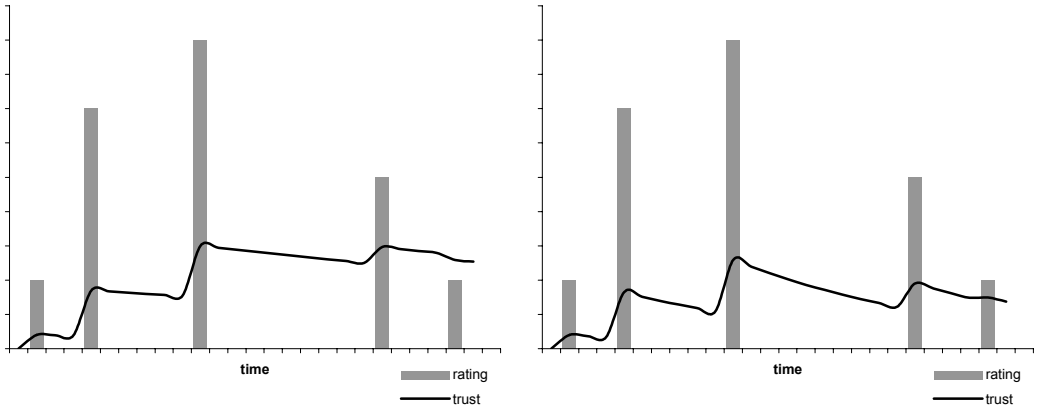
Alterung versus Abnutzung

In den Abbildungen 5.2(a) und 5.2(b) wird zusätzlich ein niedrigerer Alterungsfaktor gewählt ($\alpha = 0.2$). Hier ist zu beachten, dass einzelne Bewertungen nun einen deutlich geringeren Einfluss auf das Vertrauensverhältnis haben. Somit kommt die Gefahr auf, dass die Abnutzung den Vertrauenswert zu schnell zurück auf den Basiswert zieht. Die Abnutzungsfunktion muss daher so konzipiert sein, dass auch bei einem geringeren Alterungsfaktor die Semantik des Abnutzungsfaktors hinsichtlich der Zeiträume zwischen einzelne Bewertungen erhalten bleibt. In den Beispielabbildungen ist zu erkennen, dass sich das Vertrauen bei einem geringeren Alterungsfaktor und gleichem Abnutzungsfaktor auch langsamer abnutzt.

Bei Vertrauensberechnungsalgorithmen, die keinen Abnutzungsfaktor berücksichtigen können, weil sie den neuen Vertrauenswert bei einer Vertrauensaktualisierung nicht aufgrund des gespeicherten Vertrauenswertes berechnen, ist dieser Faktor deaktiviert und die Abnutzung komplett abgeschaltet.

5.2.3. Aktualisierungsgewicht

In den Arbeiten von [Jøsang und Ismail, 2002] und [Kinateder und Rothermel, 2003] wird beschrieben, wie Bewertungen unterschiedlich gewichtet werden können. In [Kinateder und Ro-



(a) Abnutzungsfaktor $\lambda = 0.1$ (b) Abnutzungsfaktor $\lambda = 0.4$

Abbildung 5.2.: Vertrauensverlauf mit Abnutzungsfaktor (Alterungsfaktor $\alpha = 0.2$)

thermel, 2003] ist das Gewicht ein skalarer Wert im Bereich $[0,1]$. Bei [Jøsang und Ismail, 2002] kann das Gewicht einer Aktualisierung beliebige Werte grösser als Null annehmen.

Definition 9 *Im generischen Vertrauensmodell wird das Gewicht einer Vertrauensaktualisierung i durch einen skalaren Wert* $0 \leq g_i \leq 1$ *dargestellt. Eine* bedeutungslose Aktualisierung *wird durch den Wert* 0*, eine* normale Aktualisierung *durch den Wert* 1 *beschrieben. Bei* $0 < g_i < 1$ *wird von einer* Aktualisierung mit vermindertem Gewicht *gesprochen.*

Bei [Jøsang und Ismail, 2002] sind auch Gewichtungen $g > 1$ möglich, wodurch einer Bewertung ein verstärktes Gewicht gegeben werden kann. Unsere Skala schränkt dies nicht ein, sondern stellt lediglich die Anforderung, dass die Gewichtungvariationen von Anfang an bekannt sind. Dies bedeutet z.B., dass wenn eine Gewichtung mit einem doppelten Gewicht in *beliebiger Zukunft* erwartet wird, müssen alle weiteren Gewichtungen maximal den Wert 0.5 erreichen. Alle möglichen Gewichtungen die vorkommen können, müssen somit in den Bereich $[0,1]$ normiert werden.

Das Gewicht wird bei dem Vertrauensaktualisierungsereignis einer neuen Bewertung *i* folgendermaßen berechnet:

$$g_i = gew(e_i) \cdot wert(a_i)$$

$gew(e_i)$ ist die Gewissheit des Empfehlers in seine eigene Empfehlung ($[0,1]$) und $wert(a_i)$ ist der Wert der Anfrage, die zu der Bewertung führte ($[0,1]$).

Bei anderen Ereignissen wird das Gewicht ähnlich berechnet. Wenn der Algorithmus z.B. eine Aktualisierung an semantisch verwandten Kategorien übertragen will, kann das Gewicht durch den Verwandtschaftsgrad ($g_{x,y}$) vermindert werden: $g_i \cdot g_{x,y}$.

6. Das generische Vertrauensmodell in UniTEC

Das generische Vertrauensmodell, welches im vorherigen Kapitel definiert und beschrieben wurde, wurde in das bestehende UniTEC-System integriert. Dieses Reputationssystem beruht auf verteilte Agenten, die stellvertretend für ihre Benutzer in einem Peer-to-Peer Netzwerk miteinander kommunizieren (siehe Terdic [2003]). Die Benutzer bleiben durch Pseudonyme gegenseitig anonym (siehe Widmaier [2004]).

Um diese Integration zu erläutern werden zunächst die wichtigsten Anwendungsfälle beschrieben, die bei der Vertrauensbetrachtung im UniTEC relevant sind. Anschließend wird die Architektur beschrieben, die das generische Vertrauensmodell realisiert, und die eine Integration lokaler Vertrauensmodelle möglich macht. Die Schritte zur Integration einzelner lokalen Vertrauensmodelle und -algorithmen werden erläutert. Schliesslich wird auf die Vorteile des generischen Vertrauensmodells eingegangen.

6.1. Use-Cases

In der Abbildung 6.1 werden die Anwendungsfälle beschrieben, die im UniTEC-System für die Vertrauenswertberechnung interessant sind.

Die beteiligten Rollen sind dabei:

Benutzer engl. *user*
: Eine beliebige Person, die das UniTEC-System benutzt.

Empfehler engl. *recommender*
: Eine Person in der Rolle des Urhebers einer Empfehlung.

Agent engl. *agent*
: Ein Computersystem, das im Auftrag eines Benutzers autonom Aktivitäten im UniTEC-System durchführt.

Zu den einzelnen Anwendungsfällen lassen sich folgende wichtige Merkmale in Bezug auf die Vertrauensberechnung erkennen:

Anfrage stellen engl. *create query*
: Ein Benutzer erstellt im UniTEC-System eine Anfrage und muss dabei die Wichtigkeit in

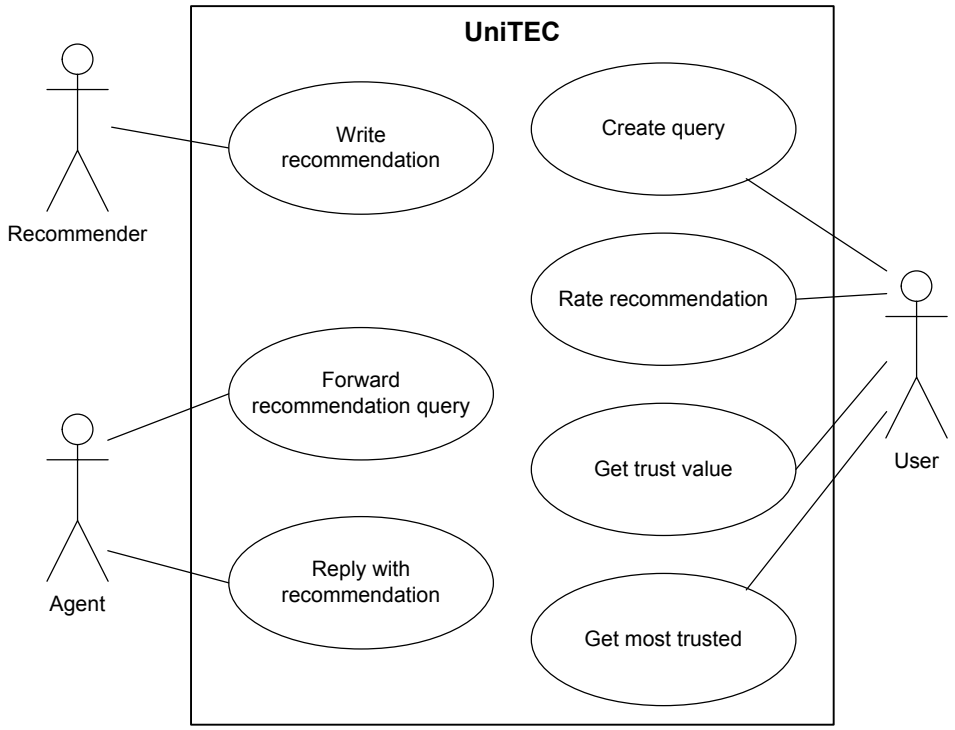


Abbildung 6.1.: Anwendungsfälle (Use Cases) für Vertrauensberechnungen bei UniTEC

Form eines Anfragewertes festlegen, die bei späteren Vertrauensberechnungen eventuell bei der Gewichtung der Bewertung berücksichtigt wird. Die Anfrage wird außerdem mit einer Menge von Kategorien in Verbindung gesetzt, was ebenfalls später Einfluss auf die Vertrauensberechnungen haben kann.

Empfehlung schreiben engl. *write recommendation*

Ein Benutzer kann Empfehlungen zu beliebigen Objekten schreiben und bei seinem Agenten abspeichern. Falls erlaubt, stehen diese dann im UniTEC-Netz anderen Benutzern zur Verfügung. Für die Vertrauensberechnungen interessant ist die Situation, in der ein Benutzer eine Empfehlung zu einem Objekt schreibt, zu dem er schon fremde Empfehlungen gespeichert hat. Falls es möglich ist die Empfehlungen inhaltlich zu vergleichen, kann sein Agent automatisch die fremden Empfehlungen bewerten. Dieses automatische Bewerten führt dann die gleichen Berechnungen durch wie der Anwendungsfall „Empfehlung bewerten“.

Um diese Situation zu veranschaulichen, stelle man sich folgenden Ablauf vor: Ein Benutzer will ein Objekt kaufen, und braucht dazu Empfehlungen. Dazu stellt er eine Anfrage an das UniTEC-System und erhält von fremden Pseudonymen entsprechende Empfehlungen. Diese kann er jedoch noch nicht konkret bewerten, weil er das Objekt an sich noch nicht kennt. Er kauft aufgrund der Empfehlungen (z.B. der vertrauenswürdigen Pseudonyme) nun das Objekt. Nach einer gewissen Zeit hat er selbst genug Erfahrungen mit dem Ob-

jekt gesammelt und kann nun selbst eine Empfehlung dazu schreiben. Diese Empfehlung wird die gespeicherten Empfehlungen zu dem Objekt bestätigen oder wiederlegen. Im ersten Fall könnte sich dadurch das Vertrauen in den Urheber der Empfehlungen erhöhen, andernfalls verringert sich das Vertrauen.

Empfehlungsanfrage weiterleiten engl. *forward recommendation query*
Ein Agent kann eine Anfrage an vertrauenswürdige Nachbarpseudonyme weiterleiten. Die Berechnung dieser Nachbarschaft kann auf Vertrauensinformationen beruhen. Außerdem muss der Agent bei der Weiterleitung die entsprechende Vertrauenskette ergänzen (mehr zu Vertrauensketten in Abschnitt 6.4.2).

Empfehlung zurücksenden engl. *reply with recommendation*
Wenn ein Agent in seiner Empfehlungsdatenbank eine passende Empfehlung findet, kann er diese an den Anfragenden zurücksenden. Davor muss er noch die Vertrauenskette endgültig mit seinem Vertrauensverhältnis zu dem Urheber der entsprechenden Empfehlung abschließen.

Empfehlung bewerten engl. *rate recommendation*
Nach Lektüre einer fremden Empfehlung kann der Anwender diese auch bewerten. Dies kann bestimmte Vertrauensberechnungen anstoßen. Dies ist das wichtigste Ereignis für Vertrauensaktualisierungen.

Vertrauenswert holen engl. *get trust value*
Der Benutzer will zu einem bestimmten fremden Anwender in einen bestimmten Kontext das aktuell herrschende Vertrauensverhältnis erfahren.

Vertrauenswürdigste holen engl. *get most trusted*
In einem bestimmten Kontext will der Benutzer die vertrauenswürdigsten fremden Benutzer aufgelistet bekommen. Auch der Agent kann diese Anfrage für eigene Berechnungen (z.B. zur Berechnung der Nachbarschaft) nutzen.

6.2. Architektur

Die Vertrauenskomponente ist wie in Abbildung 6.2 dargestellt in UniTEC integriert.

Die Vertrauenskomponente bietet zunächst anderen lokalen UniTEC-Komponenten eine einheitliche Schnittstelle an, um auf Vertrauensinformationen zugreifen oder diese beeinflussen zu können. Das generische Vertrauensmodell beschreibt die Art, wie die Informationen nach außen hin sichtbar sind. Die eigentlichen Vertrauensberechnungen werden durch Vertrauensaktualisierungsalgorithmen realisiert, die entweder direkt mit dem generischen Vertrauensmodell arbeiten können (im Beispiel, Algorithmus 3) oder mit einen spezifischen lokalen Vertrauensmodell arbeiten (Algorithmen 1 und 2). Das lokale Vertrauensmodell dient in diesem Fall als Adapter zu dem generischen Vertrauensmodell für diese Algorithmen. Die Algorithmen können zudem über entsprechenden Schnittstellen auf andere UniTEC-Komponenten zugreifen, um z.B. an der

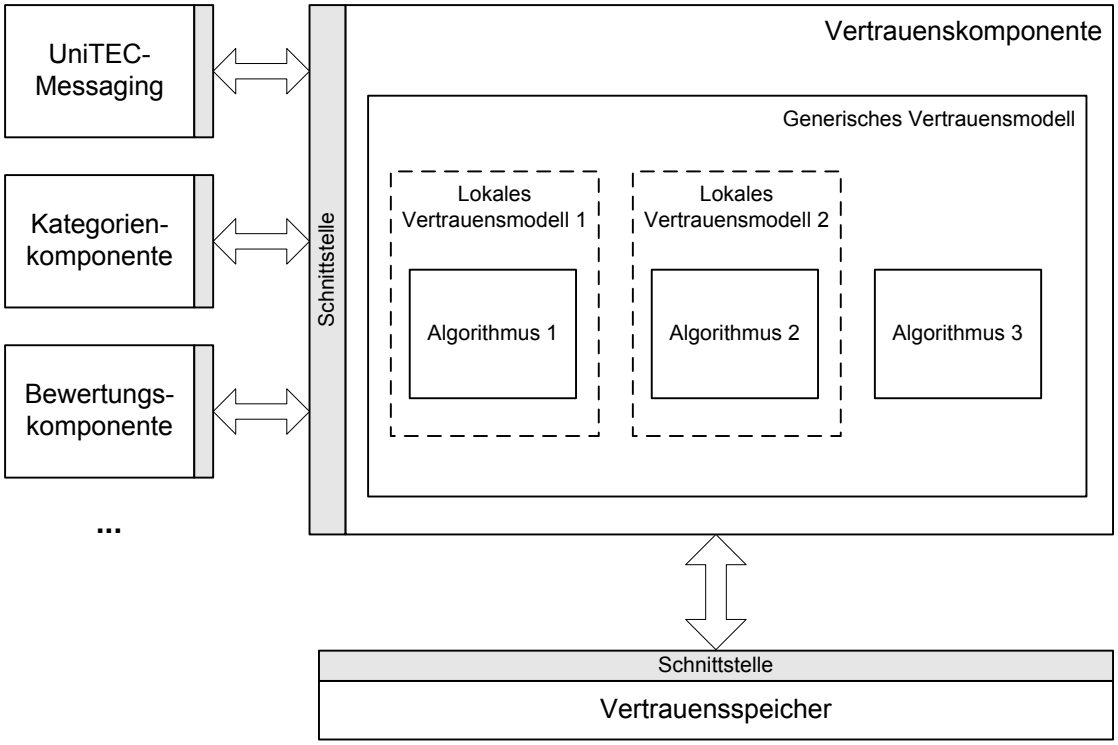


Abbildung 6.2.: Architektur der Vertrauenskomponente in UniTEC

UniTEC-Kommunikation teilzunehmen. Zur persistenten Speicherung der Vertrauensinformationen wird durchgängig die Speicherkomponente verwendet.

Der Benutzer eines Agenten wählt die Algorithmen und die Parameter für die Vertrauensberechnungen. Die Algorithmen sind auswechselbar, die Vertrauenskomponente arbeitet nach außen als eine Black-Box.

6.3. Integration lokaler Vertrauensmodelle und -algorithmen

6.3.1. Anforderungen

Um ein Vertrauensmodell und das entsprechende Vertrauensaktualisierungsalgorithmus in UniTEC umzusetzen, sind einige Kriterien zu erfüllen. Diese werden im Folgenden beschrieben.

Metriken

Bei den Metriken (Vertrauens-, Vertrauensgewissheits- und Bewertungsmetrik) sind folgende Eigenschaften notwendig:

- Die Werte müssen miteinander vergleichbar sein.
- Es müssen ein minimaler und ein maximaler Wert existieren.
- Es muss eine bijektive Abbildung dieser Metrik in die entsprechende Metrik des generischen Vertrauensmodells existieren.

Die Vertrauens- und Bewertungsmetrik sind mindestens notwendig. Eine Gewissheitsmetrik muss nur dann vorhanden sein, wenn der jeweilige Vertrauensalgorithmus eine Gewissheit berechnen kann. Wenn dies der Fall ist, muss auch diese den obigen Anforderungen entsprechen.

Vertrauensmodell

Ein Vertrauensaktualisierungsalgorithmus kann direkt mit den Metriken des generischen Vertrauensmodells arbeiten, wenn er diese direkt verstehen kann. Ansonsten muss ein lokales Vertrauensmodell (welches die beschriebenen Anforderungen der Metriken erfüllt) als Adapter erstellt werden, auf dem er arbeiten kann.

Vertrauensinitialisierung

Ein Bestandteil eines Vertrauensaktualisierungsalgorithmus ist der Initialisierungsalgorithmus, der zu einem fremden Pseudonym, zu dem noch keine Erfahrungen gemacht wurden, ein initialies Vertrauensverhältnis bestimmt.

Ereignissorientierung

Ein Vertrauensaktualisierungsalgorithmus arbeitet ereignisorientiert und ändert auf Anforderung Vertrauensverhältnisse. Auf folgende Ereignisse kann ein Vertrauensaktualisierungsalgorithmus z.B. reagieren:

- Eintreffen einer neuen Bewertung.
- Rausfall einer Gruppe von Bewertungen aus dem Bewertungsfenster („Bewertung vergessen“).
- Ein lesender Zugriff auf ein Vertrauensverhältnis. Damit hat der Vertrauensaktualisierungsalgorithmus die Möglichkeit das Vertrauensverhältnis zu ändern, bevor es verwendet wird (z.B. wenn der Alterungsfaktor berücksichtigt werden soll).
- Eintreffen neuer Empfehlungen.

Hilfsmittel

Ein Vertrauensaktualisierungsalgorithmus nutzt folgende lokale Hilfsmittel für seine Berechnungen:

- Kategorienmodell (siehe Anhang A.1)
- Bewertungshistorie (siehe Anhang A.3)
- Benutzerparameter (Alterungsfaktor, Abnutzungsfaktor, Aktualisierungsgewicht)
- Daten des Vertrauensaktualisierungsereignisses

Vertrauensmodelle, die Fremde Aussagen oder die Analyse des sozialen Netzes in die Vertrauensbetrachtungen nutzen, können dieses durch Anfragen in das Peer-to-Peer Netz realisieren. Somit kann z.B. eine „Reputationsanfrage“ an die Nachbarschaft gestellt werden, um eine Vertrauensinitialisierung durchzuführen. Hierzu mehr in Abschnitt 6.4.2.

Persistenter Speicher

Zwischenergebnisse oder sonstige Daten können durch den Algorithmus persistent gespeichert werden, um spätere Berechnungen zu beschleunigen. Der Algorithmus muss jedoch auch in der Lage sein, nur anhand der Standardhilfsmittel aktuelle Vertrauenswerte zu berechnen.

Alterung und Abnutzung

Ein Vertrauensaktualisierungsalgorithmus kann den Wert des Alterungsfaktors nach eigenem Ermessen verwenden. Lediglich die Effekte in den Extremfällen (siehe Abschnitt 3.8.1) müssen erfüllt werden. Falls ein Vertrauensaktualisierungsalgorithmus den Abnutzungsfaktor verwenden will, muss er die beschriebenen Fälle unterstützen (Abschnitt 3.9). Lediglich die Funktion, die das Abnutzen des Vertrauenswertes im Zeitraum zwischen Eintreffen des letzten Ereignisses und Ablauf der zeitlichen Grenze beschreibt, bleibt dem Algorithmus selbst überlassen.

Aktualisierungsgewicht

Jedes Ereignis, das das Vertrauen aktualisieren kann, wird mit einer Gewichtung versehen, die dessen Wichtigkeit beschreibt. Der Vertrauensaktualisierungsalgorithmus kann diesen Wert in seinen Berechnungen verwenden. Die Semantik dieses Faktors wird durch das generische Vertrauensmodell definiert (siehe Abschnitt 3.4).

6.3.2. Integrationsschritte

Die Architektur (siehe Abschnitt 6.2) ist so gestaltet, dass relativ einfach neue Vertrauensmodelle und entsprechende Vertrauensalgorithmen in das UniTEC-System integriert werden können.

Folgende Schritte sind für eine solche Integration notwendig:

1. **Vertrauensmodell**: Eventuell müssen Adapterklassen zu den Metriken und Faktoren des generischen Vertrauensmodells implementiert werden, falls das lokale Vertrauensmodell neu ist und solche noch nicht vorhanden sind.

2. **Vertrauensalgorithmus**: Der entsprechenden Vertrauensalgorithmus muss implementiert werden, und dabei vorgeschriebene Schnittstellen implementieren (siehe Anhang A.2). Alternativ dazu kann der Algorithmus völlig unabhängig von UniTEC geschrieben werden, wobei dann eine Brückenklasse mit den vorgeschriebenen Schnittstellen notwendig ist. Die Implementierung des Algorithmus kann sehr stark vereinfacht werden, indem es die durch UniTEC zur Verfügung gestellten Mittel nutzen kann: das Kategorienmodell, ein algorithmusspezifischer Datenspeicher, der generische Vertrauensspeicher, kontextsensitive Vertrauensbehandlung (d.h. unterschiedliche Vertrauensverhältnisse pro Kategorie und Vertrauensziele werden getrennt behandelt), die Bewertungsschnittstelle, das Peer-to-Peer Nachrichtensystem, die anonyme Pseudonymbehandlung, usw.

3. **Integration in UniTEC**: Die Klasse zu dem Vertrauensalgorithmus kann sich im UniTEC-System für bestimmte Ereignisse anmelden (siehe Anhang A.2.5). Diese bekommt sie dann zugestellt und kann aufgrund dessen die Vertrauensverhältnisse berechnen.

4. **Auswahl durch Benutzer**: Der Benutzer spezifiziert lediglich in seiner Konfigurationsdatei welche Klasse seinen gewünschten Vertrauensalgorithmus implementiert.

6.4. Vorteile des generischen Vertrauensmodells

6.4.1. Unterschiedliche lokale Vertrauensmodelle

Eins der Vorteile des generischen Vertrauensmodells ist die Möglichkeit, unterschiedliche lokale Vertrauensmodelle im gleichen Reputationsnetz einzusetzten. Dies bedeutet, dass jeder einzelne Benutzer selbst entscheiden kann, welches Vertrauensmodell und -algorithmus am besten zu seinem Vertrauensverständnis passt. Sofern das lokale Vertrauensmodell die Anforderungen aus Abschnitt 6.3.1 erfüllt, kann das UniTEC-System damit arbeiten.

6.4.2. Einheitliche globale Metrik

Trotz den unterschiedlich gewählten lokalen Vertrauensmodellen können Agenten miteinander kommunizieren und Vertrauensinformationen austauschen, und zwar mit den Metriken des gene-

rischen Vertrauensmodells. Vertrauensinformationen erhalten eine einheitliche Metrik und sind global verständlich und interpretierbar.

Lokale Vertrauensmodelle können eventuell Austauschmechanismen zwischen Agenten vorsehen, um Vertrauensberechnungen durchzuführen. Dies kann z.B. in Form einer Reputationsanfrage oder durch Analyse des sozialen Netzes passieren. Diese Art von Informationen ist nicht lokal verfügbar und muss daher durch Kommunikation ermittelt werden. Da jedoch jedes lokale Vertrauensmodell unterschiedliche Algorithmen dazu verwenden kann, kann dieser Austausch erstmals nur zwischen denjenigen Agenten abgewickelt werden, die auch die gleichen Algorithmen einsetzen und daher auch die Anfragen verstehen und beantworten können. Eine Vereinheitlichung einiger gängiger Kommunikationsvarianten zur Vertrauenserhebung könnte in Zukunft diese Einschränkung minimieren. Dazu spielt ebenfalls des generische Vertrauensmodell eine wichtige Rolle.

Vertrauensketten

Eine Anwendung des generischen Vertrauensmodells bei der Kommunikation stellt die Handhabung von *Vertrauensketten* dar. Bei einer Weiterleitung einer Anfrage durch das UniTEC-Netz werden die Vertrauensverhältnisse zwischen den einzelnen vermittelnden Knoten in Form von Vertrauensketten festgehalten. Mit jeder Antwort erhält der Anfragende die komplette (verifizierbare) Vertrauenskette mitgeteilt. Eine Vertrauenskette ist eine geordnete Menge von Vertrauensbeziehungen. Die Kette stellt eine transitive Beziehung dar und ergibt sich aus:

$$(p_1, p_2, T_{p_1,p_2}(k, z_w)), \ldots, (p_{n-2}, p_{n-1}, T_{p_{n-2},p_{n-1}}(k, z_w)), (p_{n-1}, p_n, T_{p_{n-1},p_n}(k, z_e))$$

Im ersten Tupel der Kette ist p_1 der Anfragende. In einer Antwortnachricht ist p_n im letzten Tupel das Pseudonym des Urhebers der Empfehlung, so dass transitiv das Vertrauen des Anfragenden in den Empfehler berechnet werden könnte, falls das Vertrauensmodell dies vorsieht. z_w beschreibt den Vertrauensziel der Weiterleitung und z_e den der Empfehlung. T stellen jeweils die einzelnen Vertrauensbeziehungen dar (Vertrauen und Gewissheit).

Diese Informationen dienen zur Auswahl und Evaluation von bevorzugten Empfehlungen oder weitere Vertrauensaktualisierungsvorgänge. Eine zukünftige Arbeit wird sich mit der Interpretation dieser Vertrauensketten beschäftigen. Da nun unterschiedliche lokale Vertrauensmodelle die Vertrauensbeziehungen lokal berechnen, dient hier ebenfalls das generische Vertrauensmodell als Metrik der Informationen in diesen Vertrauensketten. Diese sind somit global interpretierbar, unabhängig vom eingesetztem lokalen Vertrauensmodell der einzelnen Vermittlungsagenten.

6.4.3. Einheitliche lokale Metrik

Unabhängig von dem eingesetztem lokalen Vertrauensmodell basieren einige lokale Berechnungen (die z.B. Vertrauensentscheidungen hervorrufen) im UniTEC bereits auf einem Vertrauenskonzept. Bislang wurde das Vertrauensmodell aus [Kinateder und Rothermel, 2003] als Grundlage dazu verwendet. Das generische Vertrauensmodell abstrahiert von diesem Modell abstrahieren und ersetzt es zu diesem Zweck.

7. Umsetzung vorhandener Vertrauensmodelle und -algorithmen

Im Folgenden werden Details zu der Umsetzung der relevanten Vertrauensmodelle und der entsprechenden Vertrauensaktualisierungsalgorithmen in das UniTEC-System mit dem generischen Vertrauensmodell vorgestellt. Es werden, wie schon in Abschnitt 4.2, die Bezeichnungen vereinheitlicht.

Alle nachfolgend dargestellte Vertrauensberechnungen beziehen sich auf eine Vertrauensbeziehung zwischen genau zwei Agenten in genau einem bestimmten Kontext und zu einem Vertrauensziel. Durch diese Annahme werden jeweils Indizes oder Bezeichnungen für Kontext und beteiligte Agenten weggelassen, die in der Literatur auch teilweise vorkommen.

7.1. Allgemein

7.1.1. Aktualisierungstyp

Bei der Umsetzung der Aktualisierungsalgorithmen wurde zwischen Updatealgorithmen und Evolutionsalgorithmen unterschieden (siehe auch [Jonker und Treur, 1999], *update function* oder *evolution function*). Bei einem **Updatealgorithmus** basiert eine Vertrauensberechnung auf einem neuen Bewertungswert und dem zuletzt gespeicherten Vertrauenswert. Die komplette Bewertungshistorie, die dazu führte, ist dabei irrelevant. Anders, basiert ein **Evolutionsalgorithmus** seine Berechnungen auf eine Analyse der komplett zu berücksichtigenden Bewertungshistorie. Allerdings lassen sich viele Evolutionsalgorithmen so optimieren, dass nach jeder Berechnung jeweils ein Zwischenzustand gespeichert wird, auf dem die nächste Aktualisierungs beruhen kann, ohne dass wieder die komplette Bewertungshistorie berücksichtigt werden muss. Solche Algorithmen werden von hier ab nun **Pseudoupdatealgorithmen** genannt.

Update- und Evolutionsalgorithmen

Pseudoupdatealgorithmen

7.1.2. Metriken

Sämtliche Metriken des generischen Vertrauensmodells sind skalar. In vielen der Vertrauensmodelle aus der Literatur werden für einzelne Werte ebenfalls skalare Metriken verwendet. Diese (x_l) können durch entsprechende Verschiebung und Skalierung in die Metriken des generischen Vertrauensmodells (x_g) umgerechnet werden:

$$x_g = (x_l - min_l) \cdot \frac{max_g - min_g}{max_l - min_l} + min_g$$

Dabei ist min_l und max_l jeweils der minimale und maximale Wert der lokalen Metrik, min_g und max_g die minimalen und maximalen Werte der generischen Metrik. Falls die lokale Metrik eine gegenseitige Semantik hat, wie im generischem Vertrauensmodell (z.B. es wird die Ungewissheit statt Gewissheit in einem skalaren Bereich aufgetragen), muss zusätzlich eine Umpolung der Metrik stattfinden. Dieses kann bei der Berechnung durch Umtauschen von min_l und max_l umgesetzt werden.

Bei lokalen diskreten Metriken ist eine semantische Analyse der Bedeutung der einzelnen diskreten Werte erforderlich, um daraus sinnvolle Werte der skalaren generischen Metrik zu erzeugen.

7.2. Vertrauensmodelle

7.2.1. Jonker [1999]

Bei dem Vertrauensaktualisierungsalgorithmus handelt es sich um einen *Updatealgorithmus*, der jeweils nur auf den letzten Vertrauenswert (T_{i-1}) und der neuen Erfahrung in Form einer Bewertung (b_i) beruht und daraus den neuen Vertrauenswert (T_i) berechnet. Der Alterungsfaktor (α) wurde dabei anstelle der „inflation rate“ (d) verwendet:

$$d = 1 - \alpha$$

Gewichtung

Eine Gewichtung einzelner Bewertungen konnte in der Art integriert werden, indem zu jeder Aktualisierung das entsprechende Gewicht ebenfalls als Faktor berücksichtigt wurde. So ergab sich eine Vertrauensaktualisierung aus folgender Berechnung:

$$T_i = (1 - (\alpha \cdot g_i)) \cdot T_{i-1} + (\alpha \cdot g_i) \cdot b_i$$

7.2.2. Bin Yu [2000]

In [Yu und Singh, 2000] wird ein lokaler Vertrauensaktualisierungsalgorithmus beschrieben, der ebenfalls auf dem zuletzt gespeicherten Vertrauenswert und einer neuen Bewertung beruht, also handelt es sich ebenfalls um ein *Updatealgorithmus*.

Die Bewertungsmetrik (siehe Abschnitt 4.2.2) ist etwas komplexer aufgebaut als die entsprechende Metrik aus dem generischen Vertrauensmodell. Die Umrechnung einer Bewertung im generischen Vertrauensmodell in die Metrik von [Yu und Singh, 2000] erfordert daher einige

Bewertungsmetrik

Überlegungen . Zunächst wurde die Grenze von 0.5 zur Unterscheidung zwischen positiven und

Bewertung b	Erfahrung	Parameter für [Yu und Singh, 2000]
$0 \leq b < 0.5$	negativ	$-0.5 \leq \beta < 0$
$0.5 \leq b \leq 1$	positiv	$0 \leq \alpha \leq 0.5$

Tabelle 7.1.: Beispiel zur Konvertierung der Bewertungsmetrik von [Yu und Singh, 2000]

negativen Bewertungen verwendet. Da in diesem Vertrauensmodell jedoch trotzdem sowohl positive als auch negative Bewertungen zusätzlich in einer skalaren Metrik eingeordnet werden, geht bei der Umrechnung trotzdem keine Information verloren. Anschließend wird α bzw. β entsprechend benutzerkonfigurierbaren Parameter skaliert. Als Beispiel könnte eine Konvertierung verwendet werden, wie in Tabelle 7.1 beschrieben.

Es sind somit unabhängige Skalierungen von α und β möglich, womit die Anforderung $|\alpha| < |\beta|$ erfüllt werden kann. Allerdings ist aus der Berechnungstabelle 4.1 (Seite 33) zu erkennen, dass im dem Fall, in der einer Erfahrung mit $\alpha = 1$ bzw. $\beta = -1$ registriert wird, bei der Dynamik der Vertrauensentwicklung ein Fangzustand beim blinden Vertrauen bzw. beim absoluten Misstrauen erreicht wird, der nicht mehr verlassen werden kann. Außerdem besteht in diesen Fällen auch die Gefahr einer Teilung durch Null. Aus diesem Grund sollten diese Extremwerte bei der Konvertierungsvorschrift nicht möglich sein (also $-1 < \beta < 0$ und $0 < \alpha < 1$).

7.2.3. Rahman [2000]

Zunächst wurde die diskrete Metrik zu Vertrauen aus [Abdul-Rahman und Hailes, 2000] durch folgende ad-hoc Vorschrift in Vertrauenswerte des generischen Vertrauensmodells abgebildet: Der Bereich $[0..1]$ wurde in drei gleichmäßigen Teilbereiche aufgeteilt. Die Werte an den Grenzen der Bereiche repräsentieren die vier diskreten Werte (also $0, 1/3, 2/3, 1$). Jeder diskrete Wert wird einer eindeutigen Positionsnummer zugeordnet (konkret: $pos(0) = vu$, $pos(1) = u$, $pos(2) = t$ und $pos(3) = vt$). Folgende Vorschrift gilt, um einen Vertrauenswert des generischen Vertrauensmodells t_g in einen der vier diskreten Werte t_r aus [Abdul-Rahman und Hailes, 2000] umzurechnen:

$$t_r = pos(round(t_g \cdot 3))$$

Außerdem sind folgende Werte bei einer Bewertung möglich: *very bad* = *vb*, *bad* = *b*, *good* = *g* und *very good* = *vg*. Die Umrechnung von und zu der Metrik des generischen Vertrauensmodells beruht auf der selben Berechnungsvorschrift wie bei den Vertrauenswerten.

Zuletzt existiert nicht direkt eine Gewissheitsmetrik. Es werden lediglich drei zusätzliche Zustände definiert, die eine gewisse „Unsicherheit“ ausdrücken. Im Paper wird nicht konkret beschrieben, wie diese Zustände zu interpretieren sind. Für unsere Betrachtung haben wir diese Möglichkeiten in der Metriksumrechnungen aus Tabelle 7.2 festgehalten.

Die Begründung zu dieser Umrechnung ist folgende: Bei den vier Vertrauenswerten, die keinen Unsicherheitswert darstellen, ist kein Grad an Ungewissheit bekannt. Der Fall u^0 repräsentiert unter anderem den Initialzustand (noch keine Bewertungen vorhanden), so dass es

Vertrauenswert (Rahman [2000])	Vertrauenswert (generisch)	Gewissheitswert (generisch)
u^0	0	0
u^+	$2/3$	0.5
u^-	$1/3$	0.5
vu	0	1
u	$1/3$	1
t	$2/3$	1
vt	1	1

Tabelle 7.2.: Umsetzung der Unsicherheitswerte in das generische Vertrauensmodell

Sinn macht, die Gewissheit auf 0 zu setzen. Bei weiteren Situationen, wo dieser Zustand erreicht werden kann, fehlt eine feinere Unterscheidung, wie diese Ungewissheit zu interpretieren ist. In den Fällen u^+ und u^- ist stets eine Überzahl an positiven bzw. negativen Erfahrungen vorhanden, so dass der Vertrauenswert diese Ausprägung zeigen kann. Die Gewissheit dabei ist durch eine Unsicherheit geprägt, so dass sie einen Zwischenwert einnehmen muss (0.5). Diese Entscheidungen werden später bei der Analyse der Umsetzung (Abschnitt 8.3.2) genauer betrachtet.

Hier handelt es sich erstmals um einen **Evolutionsalgorithmus**: Es werden stets alle Erfahrungen bei der Vertrauensberechnung verwendet, es wird nicht nur vom zuletzt verfügbaren Vertrauenswert und der neuen Erfahrung ausgegangen. Allerdings kann durch das Speichern der Zähler die Berechnung in Form einer Aktualisierung stattfinden (**Pseudoupdatealgorithmus**).

7.2.4. Bin Yu [2002]

In der Arbeit [Yu und Singh, 2000] kommt nicht klar hervor, wie die lokalen Informationen zu Vertrauensentscheidungen verwendet werden können. Es wird die Berechnung der Glaubwürdigkeitsfunktion $Bel(\{V\})$ angegeben, die eine Angabe des Überzeugungsgrades beschreibt, dass dem fremden Agent vertraut werden kann. Dieses Maß verwenden wir als Vertrauensmetrik. Ausserdem wird die Ungewissheit durch die Glaubwürdigkeitsfunktion $Bel(\{V, \neg V\})$ ausgedrückt. Daraus folgern wir die Vertrauens- und Gewissheitskomponenten des Vertrauensverhältnisses:

$$\begin{aligned} T_i &= m_i(\{V\}) \\ G_i &= 1 - m_i(\{V, \neg V\}) \end{aligned}$$

Bewertungen werden in elf unterschiedlichen Bewertungsgrade ($x_k \in \{0.0, 0.1, ..., 1.0\}$) abgegeben. Diese Metrik ist zwar diskret, jedoch bereits sehr feinkörnig und kann daher ohne

bemerkbaren Informationsverlust durch entsprechende Rundungen in die Bewertungsmetrik des generischen Vertrauensmodells umgerechnet werden.

Hier handelt es sich ebenfalls um einen **Evolutionsalgorithmus**. Jedoch kann durch Speichern der Zähler der einzelnen Bewertungsgrade x_k in Form von h_{x_k} die Berechnung als eine Aktualisierung passieren (**Pseudoupdate**).

7.2.5. Jøsang [2002] (Beta Reputation System)

In [Jøsang und Ismail, 2002, 2.7] wird beschrieben, wie aus einer skalaren Bewertung b_i (die im generischen Vertrauensmodell mit dem Bereich $[0,1]$ festgelegt ist) die Werte r_i und s_i berechnet werden können. In diesen speziellen Fall gilt:

$$r_i = b_i \qquad \text{und} \qquad s_i = 1 - b_i$$

Eine Gewichtung einzelner Bewertungen kann durch eine Auswahl von $r_{i,g} + s_{i,g} \neq 1$ varriiert werden ($r_{i,g}$ bzw. $s_{i,g}$ ist r_i bzw. s_i nach der Berücksichtigung des Gewichtes). Wenn $r_{i,g} + s_{i,g} = 0$, dann hat diese Aktualisierung keine Bedeutung, ist $r_{i,g} + s_{i,g} > 1$ hat die Aktualisierung verstärktes Gewicht.

Die Faktoren $r_{i,g}$ und $s_{i,g}$ werden wie folgt berechnet:

$$r_{i,g} = g_i \cdot r_i \qquad \text{und} \qquad s_{i,g} = g_i \cdot s_i$$

Beim Akkumulieren dieser beiden Faktoren kann auch den Alterungsfaktor berücksichtigt werden. In [Jøsang und Ismail, 2002] und folgende Arbeiten hat der Alterungsfaktor eine umgekehrte Bedeutung wie im generischen Vertrauensmodell: Ein Faktor von 1 schaltet den Alterungseffekt aus, was bei dem generischen Vertrauensmodell für den Wert 0 definiert ist. Deswegen wird hier anstelle der Alterungsfaktor von Jøsang und Ismail durch $1-\alpha$ ersetzt. $r(i)$ und $s(i)$ sind nun die akkumulierten Parameter nach der Bewertung i:

$$r(i) = \sum_{j=1}^{i} r_{i,g} \cdot (1-\alpha)^{i-j} \qquad \text{und} \qquad s(i) = \sum_{j=1}^{i} s_{i,g} \cdot (1-\alpha)^{i-j}$$

Um eine konstante Laufzeit pro Aktualisierungsschritt zu ermöglichen wurde die Summenbildung durch eine rekursive Vorschrift ersetzt (Hier ist x stellvertretend für r bzw. s):

$$x(i+1) = x(i) \cdot (1-\alpha) + (x_i \cdot g_i)$$

Die Beta-Verteilung für die akkumulierten $r(i)$ und $s(i)$ nach der Bewertung i berechnet sich wie folgt:

$$f(p|r(i),s(i)) = \frac{\Gamma(r(i)+s(i)+2)}{\Gamma(r(i)+1)\cdot\Gamma(s(i)+1))} \cdot p^{r(i)} \cdot (1-p)^{s(i)}$$

Der Erwartungswert der Wahrscheinlichkeit p in dieser Verteilung wird nun als das Maß des Vertrauens angesehen und wird wie folgt berechnet:

$$T_i = \frac{r(i)+1}{r(i)+s(i)+2}$$

Begrenzung durch Alterungs- und Gewichtungsfaktor

Es wurde nach ersten Betrachtungen der Testergebnisse klar, dass der Alterungsfaktor und die Gewichtung wie in [Jøsang und Ismail, 2002] und [Jøsang u. a., 2003b] beschrieben den höchstmöglich erreichbaren Vertrauenswert begrenzen. Dies kommt daher, dass bei einem Alterungsfaktor $0 < \alpha < 1$ das Akkumulieren von r und s geometrische Reihen bilden, die sich asymptotisch einen gewissen Grenzwert annähern.

Dies Berechnungsvorschrift für r und s ergibt folgende geometrische Reihe (bei maximaler Bewertungen für r mit $r_i = 1$ und bei minimaler Bewertungen für s mit $s_i = 1$):

$$x_{max} = \sum_{i=1}^{n} g \cdot (1-\alpha)^{n-i}$$

Eine geometrische Reihe dieser Art konvertiert zu einen Wert von:

$$x_{konv} = \lim_{n\to\infty} \frac{g(1-(1-\alpha)^n)}{1-q} = \frac{g}{\alpha}$$

Damit sind die minimal und maximal möglichen Vertrauenswerte:

$$T_{max} = \frac{r_{konv}+1}{r_{konv}+2} = \frac{g+\alpha}{g+2\cdot\alpha}$$
$$T_{min} = \frac{1}{s_{konv}+2} = \frac{\alpha}{g+2\cdot\alpha}$$

Somit ist klar, dass lediglich bei $\alpha = 1$ ein Vertrauenswert von 1 überhaupt erreicht werden kann. Bei kleineren Alterungsfaktoren wird der Abstand zu dem maximalen Vertrauenswert lediglich durch Erhöhen des Gewichtes g möglich. Der Vertrauenswert konvergiert erst mit $\lim_{g\to\infty}$ gegen 1.0.

In [Jøsang u. a., 2003b] ist die Alterungsfunktion ein wenig modifiziert:

$$x_{i+1} = x_i \cdot (1-\alpha) + (x \cdot g) \cdot (1-\alpha)$$

Dieses kann ebenfalls als eine geometrische Reihe aufgefasst werden:

$$x_{konv} = \lim_{n \to \infty} \frac{g \cdot (1-(1-\alpha)^n)}{1-q} - g = \frac{g}{\alpha} - g$$

Und damit folgende Grenzen der Vertrauenswerte:

$$T_{max} = \frac{g(1-\alpha)+\alpha}{g(1-\alpha)+2\alpha}$$
$$T_{min} = \frac{g(1-\alpha)+\alpha}{g(1-\alpha)+2\alpha}$$

Eine potentielle Lösung für diese Problematik ist, die Vertrauensmetrik in Abhängigkeit mit dem gewählten Alterungsfaktor zu bringen. Ausgehend von der Annahme, dass der Alterungsfaktor in einem bestimmten Kontext im Laufe einer Bewertungshistorie konstant bleibt, kann die Vertrauensmetrik so skaliert werden, dass die minimal und maximal erreichbaren Vertrauenswerte in den Bereich $[0..1]$ umgewandelt werden. Diese Alternative haben wir auch in unseren Testreihen getestet und bewertet (siehe Abschnitt 8.3.4).

Gewissheit

Zur Berechnung der Gewissheit der Vertrauensaussage werden die Gedanken aus [Jøsang und McAnally, 2004] verfolgt, wo die Ungewissheit durch die Varianz der Beta-Verteilung ausdrückt wird. Um diesen Wert normiert in unserer skalaren generischen Gewissheitsmetrik darstellen zu können, wird die Standardabweichung der Beta Verteilung verwendet. Diese berechnet sich wie folgt:

$$\sigma = \sqrt{\frac{(r(i)+1)(s(i)+1)}{(r(i)+s(i)+2)^2(r(i)+s(i)+3)}}$$

Die maximale Standardabweichung ist bei $r = s = 0$ gegeben und beträgt $\sigma_{max} = \sqrt{1/12}$. Die minimale Standardabweichung beträgt im Normalfall 0. Jedoch wird die Standardabweichung – ähnlich wie das Vertrauen – wird durch den Alterungsfaktor begrenzt, da diese ebenfalls auf die akkumulierten Werte r und s aufbaut. Auch hier kann eine Alternativskalierung der Gewissheitsmetrik auf den möglichen Bereich der Standardabweichung verwendet werden. Die minimale Standardabweichung lautet:

$$\sigma_{min} = \begin{cases} \sqrt{\frac{r_{konv}+1}{(r_{konv}+2)^2(r_{konv}+3)}} & \text{wenn} \quad \lim r = r_{konv} \\ 0 & \text{wenn} \quad \lim r = \infty \end{cases}$$

Durch diese Grenzen der Standardabweichung kann nun die Gewissheit skaliert werden, so dass die maximale Standardabweichung einen Gewissheitswert von 0 und die minimale Standardabweichung die Gewissheit von 1 repräsentieren. In den Testreihen aus Abschnitt 8.3.4 wird diese unterschiedliche Skalierung bei der Gewissheit ebenfalls deutlich.

Algorithmustyp

Auch hier handelt es sich um einen **Evolutionsalgorithmus**. Die Berechnungen beruhen jeweils auf der gesamte Bewertungsgeschichte und können nicht aus einer Kombination des letzten Vertrauenswert mit einer neuen Erfahrung bestimmt werden. Jedoch kann die Berechnung in Form einer Aktualisierung ausgeführt werden, indem die akkumulierten $r(i)$ und $s(i)$ Werte beibehaltet werden und durch die neue Bewertung lediglich aktualisiert werden. Der Alterungsfaktor kann durch die vorgestellte rekursive Vorschrift integriert werden. Somit kann das Beta Reputation System in Form eines **Pseudoupdatealgorithmus** realisiert werden.

7.2.6. Sabater [2003] (ReGreT)

Die Metriken bei [Sabater, 2003] sind bereits in skalaren Metriken angegeben, so dass die einführende Betrachtungen zu skalaren Metriken für dieses Modell angewandt werden können.

In die beispielhafte Alterungsfunktion zur Gewichtung aus Sabater kann zusätzlich eine Gewichtung der einzelnen Bewertungen eingebaut werden, indem in der Funktion $f(t_i,t)$ zusätzlich das Gewicht berücksichtigt wird:

$$f(t_i,t,g_i) = f(t_i,t) \cdot g_i$$

Wieder handelt es sich hier um ein **Evolutionsalgorithmus**. Die Berechnungen beruhen jeweils auf der gesamten Bewertungsgeschichte und können nicht aus einer Kombination des letzten Vertrauenswertes mit einer neuen Erfahrung bestimmt werden. Allerdings lässt sich mit der angegebenen Alterungsfunktion daraus kein Pseudoupdatealgorithmus erstellen.

7.2.7. UniTEC [2003]

Die Berechnungen beruhen hier wie bei [Jonker und Treur, 1999] auf ein geometrisches Lernen, wobei hier die Gewichtung explizit eine Rolle spielt (diese bildet sich aus semantischer Distanz der Kategorie und der Empfehlergewissheit). Außerdem sind die Metriken der Bewertungen und des Vertrauens anders als bei Jonker und Treur gewählt, was jedoch die Funktionsweise des

Algorithmus nicht beeinflusst. Somit wurden die Ideen aus [Kinateder und Rothermel, 2003] und [Jonker und Treur, 1999] in einem Algorithmus zusammengefügt. Es sind skalare Bewertungen möglich (vgl. Jonker und Treur [1999]), und die Bewertungen können entsprechend gewichtet werden (vgl. Kinateder und Rothermel [2003]).

Die Gewissheit, wie in [Kinateder und Rothermel, 2003] beschrieben, ist über mehrere Werte (Anzahl direkter und indirekter Erfahrungen) dargestellt, es gibt keine einheitliche Metrik. Ähnlich wie bei ReGreT kann hier eine Anzahl festgelegt, bei der der betrachtete Fremde als „Bekannt“ betrachtet wird (dort *itm*, *intimate level of interactions*). Die Gewissheit kann die gleiche Berechnungsweise wie bei ReGreT verfolgen, wo der Bekannheitsgrad und die Varianz der Bewertungen in die Gewissheitsaussage mit einfliessen. In diesem Fall errechnen wir stattdessen die Standardabweichung um die Metrik beizubehalten, gewichten dabei die einzelnen Abweichungen mit der entsprechenden Gewichtung und berücksichtigen auch die Alterung:

$$\sigma_n = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(b_i - T_n)\cdot g_i \cdot \alpha^{n-i}}$$

Hier stellt g_i die Gewichtung der Bewertung b_i dar, T_n ist der aktuelle Vertrauenswert, mit dem sämtliche Bewertungen verglichen werden. Da jede Vertrauensberechnung einen neuen Vertrauenswert ergibt, muss die komplette Bewertungshistorie für diese Gewissheitsberechnung erneut iteriert werden. Somit wird aus dem ursprünglichen **Updatealgorithmus** ein **Evolutionsalgorithmus**. Eine Evaluation dieser Gewissheitsberechnung ist im Abschnitt 8.3.7 zu finden.

7.3. Erkenntnisse bei der Umsetzung

Während der Umsetzung wurden einige Eigenschaften der Algorithmen betrachtet, mit denen teilweise in [Jonker und Treur, 1999] Vertrauensaktualisierungsfunktionen charakterisiert wurden. Einige Erkenntnisse zu den Grundfunktionalität der Algorithmen ergaben sich aus der Analyse der Forschungsarbeiten oder während dessen Umsetzung in UniTEC. Diese werden hier ebenfalls zusammengefasst.

7.3.1. Grundeigenschaften

Folgende Fragen lassen sich durch Analyse der Literatur zu den einzelnen Vertrauensmodellen beantworten.

1. Welcher initiale Vertrauenswert wird einem fremden Pseudonym, zu dem noch keine Erfahrungen bekannt sind, zugeordnet? (vgl. Jonker und Treur [1999] *maximal/minimal initial trust*)
2. Um was für ein Aktualisierungstyp handelt es sich bei dem Aktualisierungsalgorithmus? (vgl. Jonker und Treur [1999] *update function* oder *evolution function*)

3. Ist die Vertrauensdynamik monoton, d.h. gilt bei zwei Bewertungssequenzen bs_1 und bs_2 mit $bs_1 \leq bs_2$ auch $T_1 \leq T_2$? (vgl. Jonker und Treur [1999] *monotonicity*)

4. Existieren „Fangzustände“? (siehe Abschnitt 2.4 und Jonker und Treur [1999] *negative/positive fixation*)

Die Punkte 3 und 4 lassen sich bei allen analysierten Vertrauensalgorithmen auf gleicher weise beantworten. Es ergeben sich monotone Vertrauensdynamiken hinsichtlich den Bewertungssequenzen. Außerdem sind keine Fangzustände vorhanden: Ein erreichter Vertrauenszustand kann nach einer ausreichend großer Folge von bestimmten Bewertungen wieder verlassen werden.

Ansonsten sind die unterschiedlichen initialen Vertrauenswerte (Punkt 1) und die entsprechenden Aktualisierungstypen (Punkt 2) in der Tabelle 7.3 zusammengefasst.

Modell	**Initialer Vertrauenswert**	**Aktualisierungstyp**
Jonker [1999]	0 [a]	Update
Bin Yu [2000]	0	Update
Rahman [2000]	u^0 Alle Zähler $= 0$	Evolution[b]
Bin Yu [2002]	0 (mit Gewissheit: 0)	Evolution[b]
Beta Reputation System	0.5 Bei $r = 0$ und $s = 0$	Evolution[b]
ReGreT	0 [a]	Evolution
Lik Mui [2003]	0.5 Bei $p = 0$ und $n = 0$	Evolution[b]
UniTEC [2003]	0	Update

Tabelle 7.3.: Grundeigenschaften der Vertrauensalgorithmen

[a] In der Literatur wird diese Eigenschaft nicht explizit erwähnt. Dies sind für dieses Modell sinnvolle Werte.

[b] Die Aktualisierung lässt sich in Form eines **Pseudoupdatealgorithmus** realisieren.

7.3.2. Grundfunktionalität

Es werden nun einzelne allgemeine Aussagen zu wichtigen Aspekten bei der Umsetzung der Vertrauensmodelle und -algorithmen analysiert.

Aufwand

> Mit welchem Speicher- und Zeitaufwand ist die Berechnung eines Vertrauenswertes aufgrund einer neuen Bewertung zu bewältigen?

Alle Vertrauensalgorithmen vom Aktualisierungstyp **Update** benötigen während einer Aktualisierung lediglich den aktuellsten Vertrauenswert und den neuen Bewertungswert. Daraus kann ein neuer Vertrauenswert berechnet werden. Der Speicheraufwand, wie auch der Zeitaufwand, beträgt für jede Aktualisierung $O(1)$.

Bei den meisten **Evolutionsalgorithmen** kann die Vertrauensaktualisierung als eine Art **Update** implementiert werden, indem der aktuelle Zustand zwischengespeichert wird (in Form von Zählern). Danach wird inkrementell gearbeitet. Diese Variante von Evolutionsalgorithmen wird ab hier **Pseudoupdate** genannt. Es ergibt ebenfalls einen Zeit- und Platzaufwand von $O(1)$ für jede Aktualisierungsberechnung.

Bei *ReGreT*, wenn die beispielhafte Alterungsfunktion aus [Sabater, 2003] verwendet wird, ist keine Update-Variante möglich. Bei dem UniTEC-Algorithmus mit der Erweiterung zur Berechnung der Gewissheit über die Standardabweichung (siehe Abschnitt 7.2.7) kann auch kein reiner Updatealgorithmus erstellt werden. In beiden Fällen bedeutet dies, es muss bei jeder neuen Bewertung durch die komplette Bewertungshistorie iteriert werden, was einen Aufwand von $O(n)$ ergibt, wobei n die Länge der Bewertungshistorie darstellt.

Bei einem **Updatealgorithmus** bedeutet ein Zeit- und Speicheraufwand von $O(n)$, dass die kompletten vergangenen Aktualisierungsvorgänge bei jeder Aktualisierung wiederholt werden müssen, was den Aufwandsvorteil der inkrementellen Aktualisierung zunichte macht.

Konsistenz

> Ist das Vertrauensverhalten nach zwei bestimmten Zeitpunkten, wo der Vertrauenswert und die aktuelle Bewertung gleich sind, die Bewertungshistorienlänge jedoch unterschiedlich, gleich oder unterschiedlich?

Bei allen Algorithmen vom Typ **Update** ist ein neuer Bewertungswert lediglich von dem aktuellsten Vertrauenswert und dem neuen Bewertungswert abhängig, unabhängig also von der gesamten Bewertungshistorie.

Bei den **Pseudoupdate-** und **Evolutionalgorithmen** ist ein neuer Vertrauenswert von dem aktuellen algorithmusspezifischen Zustand und dem neuen Bewertungswert abhängig. Ein gleicher Vertrauenswert kann jedoch aus unterschiedlichen Zuständen hervorgehen. Daher ergibt sich nicht zwingend zu einem gleichen Vertrauenswert und einer neuen Bewertung bei einer Aktualisierung der gleiche Vertrauenswert.

Alterung

> Kann/muss ein Alterungsfaktor verwendet werden, und wie wirkt er sich aus?

Bei Jonker und UniTEC ist der Alterungsfaktor zwingender Bestandteil der Algorithmen und kann nicht ausgeschaltet werden. Für das Beta Reputation System wird in [Jøsang und Ismail,

2002] der Alterungsfaktor als optionaler Parameter beschrieben, der mit einem Wert von 0 ausgeschaltet werden kann. Dieser Alterungsfaktor kann auf gleiche Weise in den Algorithmus von [Mui, 2003] integriert werden. Bei [Sabater, 2003] wird für ReGreT eine Alterungsfunktion beschrieben, die nicht auf einen Alterungsfaktor beruht. Die Alterungsfunktion ist jedoch austauschbar und könnte in nachfolgenden Arbeiten auch einen Alterungsfaktor berücksichtigen. In [Yu und Singh, 2000] ist die Alterung implizit in den Skalierungen der Metriken für die Parameter α und β integriert. Somit kann eine flexible Alterung realisiert werden, wo gute Erfahrungen schneller veralten als schlechte Erfahrungen. In den Testreihen wird diese flexible Alterung noch nicht mit dem Alterungsfaktor gekoppelt. Es werden drei exemplarische Ausprägungen getestet: gleichstarke Alterung sowohl positiver als auch negativer Erfahrungen, stärkere Alterung von schlechten Erfahrungen und stärkere Alterung von guten Erfahrungen.

Ein wichtiger Aspekt, der bei dem Entwurf des Algorithmus bedacht werden muss, ist die zeitgleiche Nutzung der Alterung und Bewertungszeitfenster. Um bei Update und Pseudoupdate Algorithmen den Aufwand nicht zu erhöhen, muss der zwischengespeicherte Zustand zusätzliche Informationen zu der Zusammensetzung der Zähler (oder des Vertrauenswertes) halten, so dass die älteste Bewertung, trotz Alterung, aus den Betrachtungen entfernt werden kann.

Gewichtung

> Kann das Vertrauensmodell Bewertungen unterschiedlich gewichten? Wie wirkt sich die Gewichtung aus?

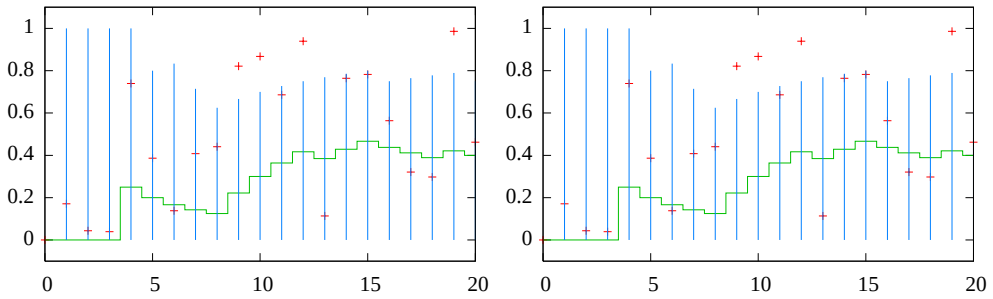
Ziel einer schwächeren Gewichtung einer individuellen Bewertung ist es, einen schächeren Einfluss auf das Vertrauensverhältnis zu verursachen. Eine Gewichtung der Bewertungen wird bereits in [Kinateder und Rothermel, 2003] für den UniTEC-Algorithmus und in [Jøsang und Ismail, 2002] für das Beta Reputation Systems beschrieben. Bei ReGreT konnte eine Gewichtung zusätzlich integriert werden. In den drei Fällen wurde das Ziel erreicht.

In zwei weiteren Algorithmen (Rahman, Bin Yu [2002]) konnte eine Gewichtung integriert werden, die jedoch die Zielsetzung nicht befriedigend erfüllte. Es wurde statt einer absoluten Gewichtung nur eine relative Gewichtung erreicht.

Bei einer *relativen Gewichtung* werden Gewichtungen miteinander verglichen: Der absolute Gewichtungswert hat keine direkte Auswirkung auf den Einfluss einzelner Bewertungen auf das Vertrauensverhältnis. In der Praxis bedeutet eine relative Gewichtung, dass bei gleichen Startbedingungen und einer gleichen Bewertungsfolge mit unterschiedlichen (gleichbleibenden!) Gewichtungen die Vertrauensdynamik identisch ist. Erst wenn unterschiedliche Gewichtungen in einer Bewertungsfolge verwendet werden, ergibt sich ein Unterschied an Vertrauensdynamiken. Bei der *absoluten Gewichtung* hingegen beeinflusst schon jede einzelne gewichtete Bewertung die Vertrauensdynamik.

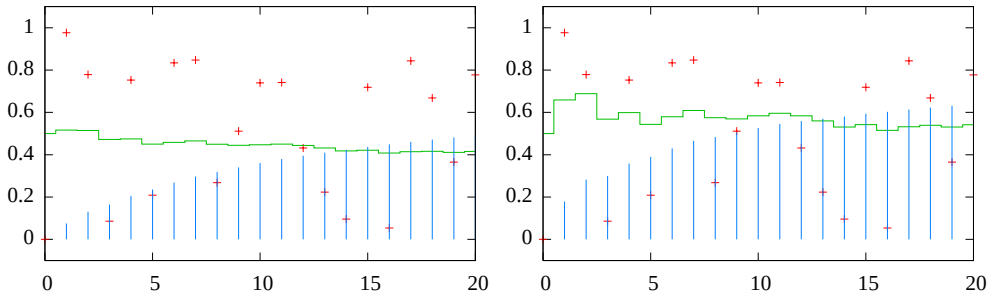
Ein Beispiel, der den Unterschied veranschaulicht, liefert ein Vergleich von Vertrauensdynamiken aus Testreihen (Abbildungen 7.1 und 7.2, diese graphische Darstellung wird auf Seite 69 beschrieben). In jeden dieser Testreihen wurden exakt die gleichen Bewertungen abgegeben.

Bei den Testreihen (a) wurde alle Bewertungen stets mit einer Gewichtung von $g = 0.3$ versehen, bei den Testreihen (b) $g = 1.0$. In Abbildung 7.1 sieht man die Auswirkung der relativen Gewichtung: Relativ zueinander sind in einer Testreihe alle Gewichtungen gleichgross, so dass die Vertrauensdynamik in Abbildungen 7.1(a) und 7.1(b) genau gleich ablaufen. Im Gegensatz dazu ist in Abbildung 7.2 der Effekt der absoluten Gewichtung dargestellt. Hier wird der Unterschied der Vertrauensdynamik bei Gewichtung $g = 0.3$ in Abbildung 7.2(a) zu der Dynamik mit Gewichtung $g = 1.0$ in Abbildung 7.2(b) schon nach jeder Bewertung deutlich.



(a) Gewichtung $g = 0.3$ (b) Gewichtung $g = 1.0$

Abbildung 7.1.: Vertrauens- und Gewissheitsdynamik bei Bin Yu [2002] im Testszenario RandomRatings (relative Gewichtung)



(a) Gewichtung $g = 0.3$ (b) Gewichtung $g = 1.0$

Abbildung 7.2.: Vertrauens- und Gewissheitsdynamik bei dem Beta Reputation System im Testszenario RandomRatings (absolute Gewichtung)

Zeitfenster

> Ist ein Zeitfenster von Bewertungen möglich, oder müssen stets alle Bewertungen für die Berechnung des Vertrauenswertes mit einbezogen werden? (vgl. Jonker und Treur [1999] *degree of memory based on window n back - forgetting about the past*)

Die meisten analysierten Algorithmen (alle mit Ausnahme von ReGreT) sind entweder vom Aktualisierungstyp **Update** oder **Pseudoupdate**. Das heisst, dass nicht die komplette Bewer-

tungshistorie berücksichtigt wird, sondern lediglich ein davon abgeleiteter Zwischenzustand. Dies hat zur Folge, dass Änderungen in der Bewertungshistorie (z.B. das „Vergessen“ der ältesten Bewertungen) keinen direkten Einfluss auf die Berechnungen haben. Daher muss dieses „Vergessen“ aktiv den Algorithmen mitgeteilt werden. Es muss dem Algorithmus dann möglich sein, aus dem aktuellen Zwischenstand (bei Pseudoupdate-Algorithmen) oder aus dem aktuellen Vertrauenswert (bei Update-Algorithmen) selektiv die betroffene Bewertung aus den Betrachtungen zu entfernen.

Abnutzung

> Existiert die Möglichkeit in dem Vertrauensmodell einen Algorithmus zur Abnutzung des Vertrauens mit Hilfe eines Abnutzungsfaktors zu integrieren?

Bei der Abnutzung soll der Vertrauenswert dynamisch bei Bedarf berechnet werden können. Dadurch soll der zuletzt durch eine Aktualisierung durch Bewertung berechnete Vertrauenswert durch einen Abnutzungsfaktor abgeschwächt werden. Bei Algorithmen vom Aktualisierungstyp **Update** bereitet diese Vorgehensweise keinerlei Probleme. Hier beruht die Berechung ohnehin auf dem zuletzt bekannten Vertrauenswert. Dieser kann auch der abgenutzte Vertrauenswert sein, der Aktualisierungsalgorithmus bleibt dadurch unverändert.

Schwieriger hingegen sieht es bei Algorithmen vom Aktualisierungstyp **Pseudoupdate** aus. Um den Aufwand einer Aktualisierung bei $O(1)$ zu belassen muss sich der Algorithmus Zwischenzustände merken. Eine Abnutzung des Vertrauenswertes kann bei der Abfrage zwar eingebaut werden, jedoch hat diese Abnutzung für den Algorithmus keinerlei Bedeutung: Die Berechnungen bei der nächsten Aktualisierung beruhen auf den Zwischenzustand, und nicht auf dem abgenutzten Vertrauenswert. Lösung dafür ist es, auch den algorithmusspezifischen Zwischenzustand abzunutzen. Wenn dieses erreicht werden kann, kann auch der Abnutzungsfaktor integriert werden.

Exemplarisch wurde die Abnutzung im Update-Algorithmus von UniTEC implementiert und getestet.

8. Analyse

Für die vergleichende Untersuchungen beschränken wir uns lediglich auf die Vertrauensberechnungen die nur mit lokalen Kenntnissen bestimmt werden können, da diese Information auf einheitliche Weise zur Verfügung gestellt werden kann (Erfahrungen in Form von Bewertungen) und in allen relevanten Modellen berücksichtigt wird.

Im Folgenden wird jedes Vertrauensmodell und der entsprechende Vertrauensalgorithmus durch Testläufe unter realen Bedingungen genauer untersucht. Jeder Algorithmus wird mit einer Reihe von **Testszenarien** konfrontiert. Hier kann beobachtet werden, wie sich die Vertrauensdynamik in den ausgewählten Situationen verhält. Schliesslich wird ein Fazit der gesamten Beobachtungen bei der Analyse gezogen.

Graphische Darstellung In den folgenden Analysen wird durchgängig eine graphische Darstellung der Vertrauensdynamik verwendet, die hier kurz erläutert wird. In Abbildung 8.1 ist dazu ein Beispielgraph zu sehen.

Auf der **y-Achse** wird die Vertrauens- und Gewissheitsmetrik des generischen Vertrauensmodells dargestellt. Die **x-Achse** repräsentiert die zeitliche Folge der Bewertungen. Die *Kreuze* stellen einzelne Bewertungen dar. Der *abgestufte Linienverlauf* beschreibt den Verlauf des Vertrauens, die dünnen *Balken* den Verlauf der Gewissheit.

8.1. Testszenarien

Jedes der folgenden Testszenarien wurde für jeden implementierten Vertrauensalgorithmus getrennt durchlaufen. Als Eingabe diente eine statische oder dynamisch generierte Sequenz von

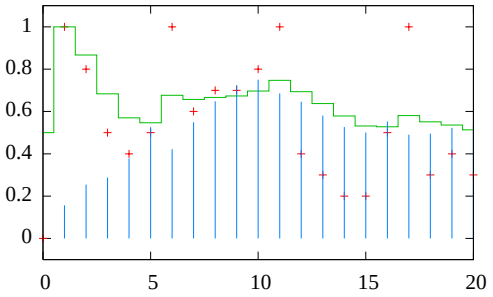


Abbildung 8.1.: Beispiel einer Vertrauensdynamik

Bewertungen. Diese Sequenz von Bewertungen wurde der Reihe nach im UniTEC System über die Bewertungskomponente gespeichert. Nach jeder gespeicherte Bewertung wurde jeweils der aktuelle Vertrauenswert abgefragt. Ausgabe war daher eine Sequenz von Vertrauenswerten.

Bei Vertrauensmodellen, die einen Alterungsfaktor verwenden, wurde der Algorithmus jeweils mit folgenden Alterungsfaktoren getestet: 0.05, 0.3, 0.5, 0.8 und 1.0. Bei Vertrauensmodellen, die den Alterungsfaktor auch gezielt ausschalten können, wurde zusätzlich der Alterungsfaktor 0.0 getestet. Jeder dieser Kombinationen stellt eine andere Ausprägung des Algorithmus dar.

Bei Vertrauensmodellen, wo Bewertungen unterschiedlich gewichtet werden können, wurde bei jeder Ausprägung des Algorithmus jedes Testszenario jeweils durch folgenden Gewichtungen abgeschwächt (zusätzlich zu der normalen 1.0 Gewichtung): 0.05, 0.3 und 0.8.

Im Folgenden werden die einzelnen Testszenarien beschrieben. In allen Szenarien wurden die Verläufe des Vertrauenswertes und der Vertrauensgewissheit betrachtet.

OnlyMaximalRatings Es werden stets maximale Bewertungen abgegeben. Wie schnell baut sich das Vertrauen auf? Gelangt es bis zu dem absoluten Vertrauen?

OnlyMinimalRatings Es werden stets minimale Bewertungen abgegeben. Wie schnell wird das Vertrauen abgebaut? Wird das totale Misstrauen erreicht?

MinimalThenMaximalRatings Nach einer Reihe von minimalen Bewertungen werden nur noch maximale Bewertungen abgegeben. Wie schnell ist ein Trendwechsel bei dem Vertrauensverhältnis zu erkennen? Wann ist das absolute Vertrauen erreicht? Wie ist diese Steigung abhängig von der Historienlänge?

MaximalThenMinimalRatings Nach einer Reihe maximaler Bewertungen werden nur noch minimale Bewertungen abgegeben. Wie schnell fällt das Vertrauensverhältnis zurück zu dem totalen Misstrauen? Wie ist der Verfall abhängig von der Historienlänge?

MaximalInterruptedByMinimal Eine lange Reihe von maximalen Bewertungen wird durch eine minimale Bewertung unterbrochen, wonach wieder maximale Bewertungen abgegeben werden. Wie ist der Einfluss dieser auf den weiteren Verlauf?

MinimalInterruptedByMaximal Eine lange Reihe von minimalen Bewertungen wird durch eine maximale Bewertung unterbrochen. Danach werden wieder minimale Bewertungen abgegeben. Wie ist der Einfluss dieser maximalen Bewertung auf den weiteren Verlauf?

SlightlyPositiveAndNegativeRatings Es werden zunächst Bewertungen knapp über (bzw. unter) der Mittelbewertung ($b = 0.5$) abgegeben, gefolgt von Bewertungen knapp unter (bzw. über) der durchnittlichen Bewertung. Kann das Vertrauensmodell mit unserer generischen Bewertungsmetrik dieses sinnvoll verarbeiten?

SpecificRatings(Normal, Reversed, OrderedAsc, OrderedDesc) Eine Menge willkürlicher Bewertungen werden in unterschiedlichen Reihenfolgen abgegeben: in einer willkürlichen Reihenfolge, in dessen umgekehrten Reihenfolge, nach Bewertungswert geordnet und schließlich umgekehrt geordnet. Sind die Bewertungen von der Reihenfolge her

austauschbar ohne den berechneten Vertrauenswert zu ändern? (vgl. Jonker und Treur [1999] *indistinguishable past*)

OscilatingRatingsFirstMinimal und -Maximal Wie verhält sich die Vertrauensdynamik bei oszillierenden Bewertungen (abwechselnd zwischen bester und schlechtester Bewertung)?

RandomRatings Wie reagiert der Algorithmus auf eine Reihe willkürlicher Bewertungen?

KeepPositive Wie oft kann ein bereits erlangtes gutes Vertrauen ($T > 0.8$) missbraucht werden (mit einer minimalen Bewertung), bis der Algorithmus das Vertrauensverhältnis als Misstrauen einordnet ($T < 0.5$)? Danach wird das Vertrauen wieder mit maximalen Bewertungen aufgebaut und es wird erneut versucht dieses zu missbrauchen.

8.2. Erwartungen

Bei den Testszenarien *SlightlyPositiveAndNegativeRatings* ist die Erwartung die, dass das Vertrauensverhältnis sich in einen durchschnittlichen Bereich einpendelt. Es soll auf keinen Fall zu Extremwerte des Vertrauens tendieren.

Eine Reihe schwach gewichteter Bewertungen beeinflusst das Vertrauensverhältnis in geringerem Maße wie die ursprüngliche Testreihe mit normaler Gewichtung.

Die Gewissheitsmetrik soll Trendänderungen bei den Bewertungen entsprechend rasch bestrafen. Dies soll in den Testszenarien *MinimalThenMaximalRatings* und *MaximalThenMinimalRatings* besonders deutlich werden.

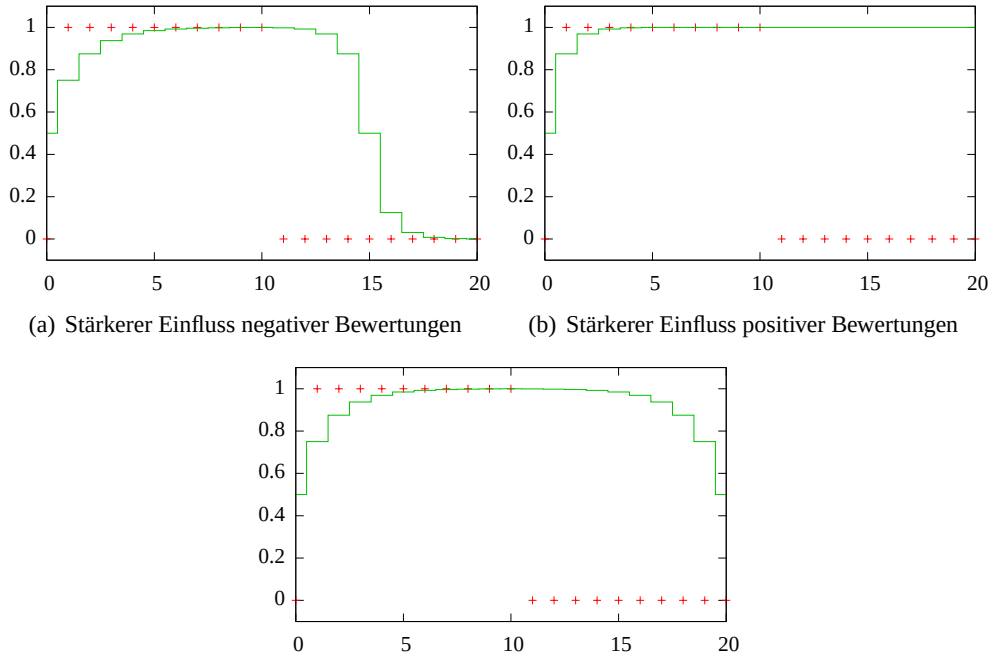
8.3. Evaluation und Ergebnisse

Die gesammelten Testszenarien und deren Ergebnisse sind ausführlich in Anhang B zu finden. Hier werden diejenigen Testergebnisse dargestellt, die genauer evaluiert werden.

8.3.1. Bin Yu [2000]

Die Vertrauensdynamik wird in diesem Modell stark durch die Skalierung der Bereiche der Parameter α und β (für positive bzw. negative Bewertungen) beeinflusst. Es sind folgende Ausprägungen möglich:

- Gleichbehandlung positiver und negativer Bewertungen
- Stärkere Gewichtung positiver Bewertungen (*pos*)
- Stärkere Gewichtung negativer Bewertungen (*neg*)



(a) Stärkerer Einfluss negativer Bewertungen (b) Stärkerer Einfluss positiver Bewertungen

(c) Gleichstarke Alterung

Abbildung 8.2.: Vertrauensdynamik bei Bin Yu [2000] im Testszenario *MaximalThenMinimalRatings*

MaximalThenMinimalRatings und *MinimalThenMaximalRatings*

In den Testszenarien *MaximalThenMinimalRatings* und *MinimalThenMaximalRatings* wird deutlich, wie sich die Dynamik unterscheiden kann. Im Normalfall ist eine stärkere Gewichtung negativer Bewertungen erwünscht. So wird in [Yu und Singh, 2000] empfohlen, dass $|\alpha| < |\beta|$. Konkret bedeutet dies, dass das Vertrauen schwer aufzubauen, aber leicht abzubauen ist. Durch diese Parametrierung ist der allgemeine Vertrauenstrend eher in negativer Richtung. In Abbildung 8.2 wird der Unterschied von drei möglichen Ausprägungen deutlich.

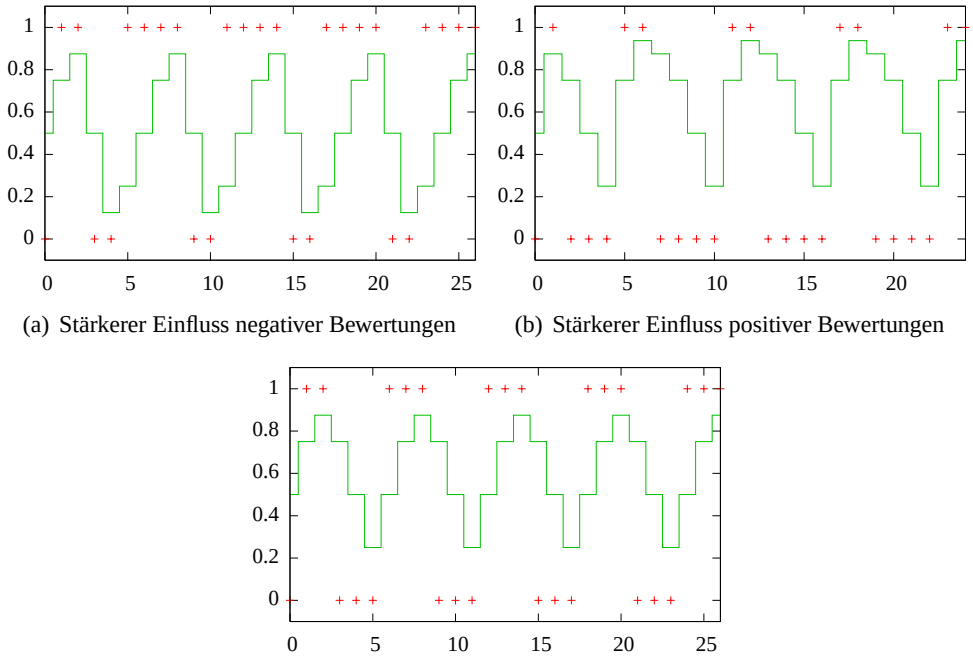
KeepPositive

Im Testszenario *KeepPositive* (siehe Abbildung 8.3) wird auch klar, dass bei stärkerer Gewichtung der negativen Bewertungen es für einen Angreifer schwerer ist, ein hohes Vertrauen zu missbrauchen, weil der Versuch relativ schnell mit einem negativen Vertrauenswert geahndet wird, während umgekehrte Parametereinstellung (stärkere Gewichtung positiver Bewertungen) den umgekehrten Effekt hat.

SpecificRatings

Da bei Bin Yu [2000] kein Alterungsfaktor möglich ist, beeinflusst die Reihenfolge der Bewertung das Endergebnis an Vertrauen nicht. Auch mit unterschiedlichen Skalierungsparameter für α und β spielt die Reihenfolge der Bewertungen für den Endwert der Bewertungen keine Rolle. Siehe dazu Abbildung 8.4.

Die Bewertungen im generischen Vertrauensmodell werden mit einem Wert aus der skalaren Bewertungsmetrik im Bereich $[0..1]$ abgegeben. Im Vertrauensmodell in [Yu und Singh, 2000] jedoch wird jede Bewertung zusätzlich zwischen „positive“ und „negative“ Bewertungtypen un-



(a) Stärkerer Einfluss negativer Bewertungen (b) Stärkerer Einfluss positiver Bewertungen

(c) Gleichstarke Alterung

Abbildung 8.3.: Vertrauensdynamik bei Bin Yu [2000] im Testszenario *KeepPositive*

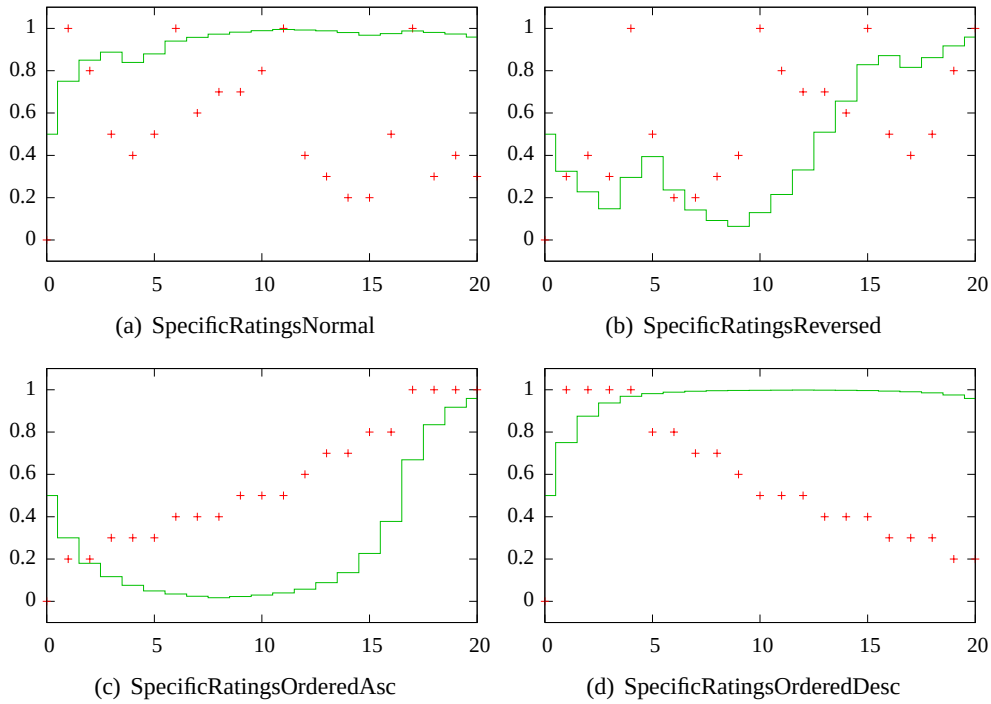
terschieden. Diese Unterscheidung wurde künstlich eingebaut, indem die Grenze von 0.5 den Bewertungstyp auseinanderhält: Bewertungen mit $b < 0.5$ wurden als negative und mit $b \geq 0.5$ als positive Bewertungen behandelt.

Allerdings verhält sich der Algorithmus bei Bewertungen, die knapp über oder unter der Durchschnittsbewertung liegen, nicht ganz den Erwartungen entsprechend. Die Vertrauensdynamik tendiert relativ schnell zu den Vertrauensextremwerten (blindes Vertrauens oder absolutes Misstrauen), auch wenn nur mäßig positive oder negative Bewertungen abgegeben wurden. In der Abbildung 8.5 wird dieses Verhalten deutlich. *SlightlyRatings*

Fazit Die Idee der unterschiedlichen Gewichtung positiver und negativer Bewertungen ist in diesen Modell ein wichtiger Aspekt und sollte in anderen Arbeiten weiter verfolgt werden. Leider wird hier kein Alterungsfaktor oder Gewichtungen berücksichtigt, auch wird keine Aussage zu der Gewissheit gemacht. Auch werden die Gedanken dieses Modells nicht mehr von Bin Yu verfolgt, siehe [Yu und Singh, 2002].

8.3.2. Rahman [2000]

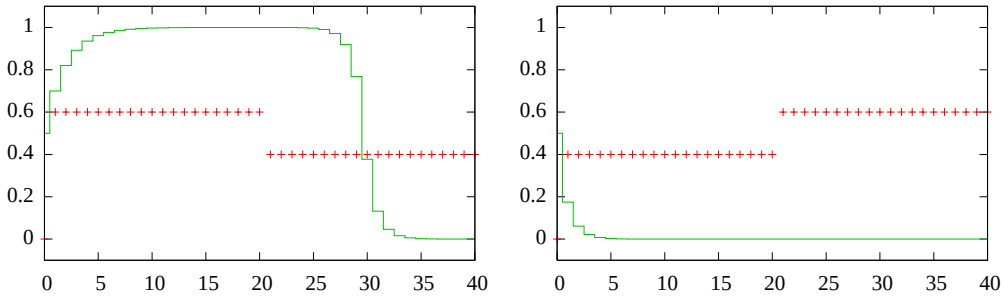
Schon nach lediglich einer ausgezeichneten Bewertung erreicht der Bewertende maximales Vertrauen, falls keine weiteren Bewertungen vorhanden sind, was in Abbildung 8.6(a) zu sehen ist. *MaximalInterrupted-ByMinimal*



(a) SpecificRatingsNormal (b) SpecificRatingsReversed

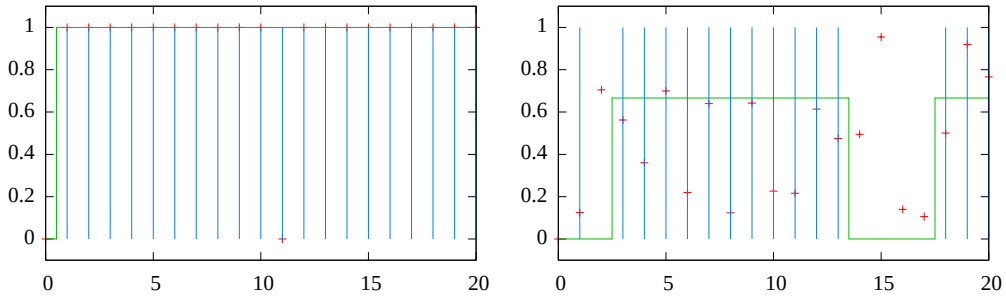
(c) SpecificRatingsOrderedAsc (d) SpecificRatingsOrderedDesc

Abbildung 8.4.: Vertrauensdynamik bei Bin Yu [2000] mit gleichstarker Behandlung positiver und negativer Bewertungen ($0 < \alpha < 0.5$ und $-0.5 < \beta < 0$)



(a) SlightlyPositiveThenNegativeRatings (b) SlightlyNegativeThenPositiveRatings

Abbildung 8.5.: Vertrauensdynamik bei Bin Yu [2000] mit stärkeren Einfluss negativer Bewertungen ($0.25 < \alpha < 0.5$ und $-0.75 < \beta < -0.5$)



(a) *MaximalInterruptedByMinimal* (b) *RandomRatings*

Abbildung 8.6.: Vertrauens- und Gewissheitsdynamik bei [Abdul-Rahman und Hailes, 2000]

Dies kommt daher, dass die Bewertungen lediglich gezählt und diese Zähler veglichen werden. Es ist für einem Angreifer relativ einfach ein aufgebautes Vertrauen beliebig oft (n Mal) zu missbrauchen, indem zuvor $n+1$ kooperative Handlungen stattfinden, die mit ausgezeichneten Bewertung belohnt werden.

Der Vertrauensalgorithmus gibt eine Vertrauensdynamik mit der vierstufigen diskreten Genauigkeit relativ gut wieder, wenn es sich nicht um Extremfälle handelt. In den Testszenarien in denen maximale und minimale Bewertungen in langen Reihenfolgen abgegeben werden, zeigt sich, dass sich der dadurch erreichte Vertrauenswert nur noch schwer beeinflussen lässt. Eine Situation mit willkürlichen Bewertungen hingegen erzeugt eine Vertrauensdynamik, die dem allgemeinen Trend mit der möglichen Genauigkeit gut verfolgt, was in Abbildung 8.6(b) zu erkennen ist. *RandomRatings*

Fazit Auch hier wird kein Alterungsfaktor verwendet. Eine Gewichtung wurde eingebaut, allerdings handelt es sich nur um eine *relative Gewichtung* (Siehe Abschnitt 7.3.2). Außerdem wird in [Abdul-Rahman und Hailes, 2000] eine dreistufige Gewissheit beschrieben, was sich nur sehr ungenau auf die skalare Gewissheit des generischen Vertrauensmodells abbilden lässt. In den meisten Fällen ist die Gewissheit maximal, da einer der Zähler herrausragt. Nur in den seltensten Fällen sind zwei oder mehrere Zähler im Gleichstand, was einen mittleren oder minimalen Gewissheitszustand entsprechen würde. Es gibt auch keine schnelle Reaktion auf einen möglichen Missbrauch eines durch mehrere gute Bewertungen aufgebautes Vertrauens. Es müssen zuerst alle die guten Bewertungen „aufgebraucht“ werden, bevor das Vertrauen sinkt.

Bei der Abbildung der skalaren Metriken des generischen Vertrauensmodells in die grobkörnige diskreten Metrik ist natürlich mit Informationsverlust behaftet. Eine sinnvolle Bereichsbildung zur Abbildung ist dazu notwendig, die die diskrete Metrik aus den sprachlichen Konstrukten möglichst präzise in der skalaren Metrik abbildet. Bei [Abdul-Rahman und Hailes, 2000] werden z.B. keine Aussagen gemacht, was genau bei den Unsicherheitswerten zu verstehen ist. Hier wurde eine Abbildung wie in Abschnitt 7.2.3 beschrieben gewählt. Es wird klar, dass keine genaue Umsetzung der Gewissheit möglich ist, da diese nicht ausreichend im Ursprungsvertrauensmodell beschrieben wird. Die naive Umsetzung mit den Ausprägungen der Gewissheit von

0, 0.5 und 1 erweist sich zu schwach: nach einer Bewertung kann die Gewissheit einen Sprung von 0 auf 1 machen, was natürlich nicht Sinnvoll ist. Schlussfolgerung daraus ist, dass die bei Rahman genannten Unsicherheitswerte unzureichend interpretierbar sind, und daher eine besser Gewissheitsmetrik eingesetzt werden sollte. Eine Möglichkeit ist eine Gewissheitsberechnungen wie bei ReGreT oder bei der Erweiterung im UniTEC-Algorithmus (siehe Abschnitt 7.2.7): Einen Faktor abhängig von der Anzahl an Begegnungen und ein weiteren Faktor aus der Varianz der Begegnungen zu dem aktuellen Vertrauenswert.

8.3.3. Bin Yu [2002]

Für unsere Tests wurde der untere Schwellwert auf $\omega = 0.4$ und der obere Schwellwert auf $\Omega = 0.6$ gesetzt. Bewertungen $b > 0.6$ trugen somit zu dem Vertrauensaufbau, $b < 0.4$ zu einem Vertrauensabbau und $0.4 \leq b \leq 0.6$ zu einer Erhöhung der Unsicherheit bei.

Für die Umrechnung in das generische Vertrauensmodell wurde der Vertrauenswert direkt aus der *belief function* $Bel(\{V\})$ abgeleitet (siehe Abschnitt 7.2.4). Außerdem wurde die Gewissheit aus der *belief function* $Bel(\{V, \neg V\})$ hergeleitet, was nur bedingt zu guten Ergebnissen führt (hier wird die relative Häufigkeit von Bewertungen im mittleren Bereich als Gewissheitskriterium verwendet). Die Aufteilung in den Bereichen (durch Auswahl der beiden Schwellwerten ω und Ω) erlaubt in dieser Richtung eine Feineinstellung, ändert aber am Grundprinzip nichts. Weitere Möglichkeiten eine Gewissheit aus den Eingangsdaten zu errechnen wurden nicht gegeben. Deutlich ist dieser Mangel in allen Testszenarien aus Abbildung 8.7 zu sehen: Da hier nur extreme (maximale oder minimale) Bewertungen abgegeben werden, werden nie Bewertungen in dem Ungewissheitsbereich (zwischen $\omega = 0.4$ und $\Omega = 0.6$) abgegeben. Somit ergibt sich stets die minimalste Ungewissheit, was als eine maximale Gewissheit bewertet wird.

OnlyMaximalRatings

Bei den Testszenarien stellt sich heraus, dass schon nach lediglich einer maximalen Bewertung, ein neuer Bewertender das maximale Vertrauen erreicht, was in Abbildung 8.7(a) zu sehen ist.

MaximalThen-MinimalRatings, KeepPositive, MaximalInterrupted-ByMinimal

Die Eigenschaft der relativen Häufigkeit verhindert zudem eine schnelle Reaktion auf einen Missbrauch eines durch mehrere guten Bewertungen aufgebautes Vertrauen, wie in Abbildung 8.7(b) oder auch in Abbildung 8.7(c) deutlich wird. In 8.7(d) ist umgekehrt zu erkennen, wie eine einzige minimale Bewertung eine lange Reihe maximaler Bewertungen starken Einfluss auf das Vertrauensverhältnis hat.

Fazit Hier wird ebenfalls kein Alterungsfaktor verwendet. Eine Gewichtung der Bewertungen kann hinzugefügt werden, indem die einzelne Bewertungen gewichtet in der Berechnung der relativen Häufigkeit berücksichtigt werden. Allerdings handelt es sich auch hier, wie schon bei [Abdul-Rahman und Hailes, 2000], um eine *relative Gewichtung* (siehe dazu Abschnitt 7.3.2), da auch hier die Zähler der drei Bereiche der *bpa* relativ zueinander berücksichtigt werden, und der absolute Wert keine Bedeutung hat.

Die Lösung in [Yu und Singh, 2002] beruht auf Erkenntnisse aus dem Dempster-Shafer Kalkül, erfordert jedoch noch weitere Schritte für einen praktischen Einsatz was die lokale Ver-

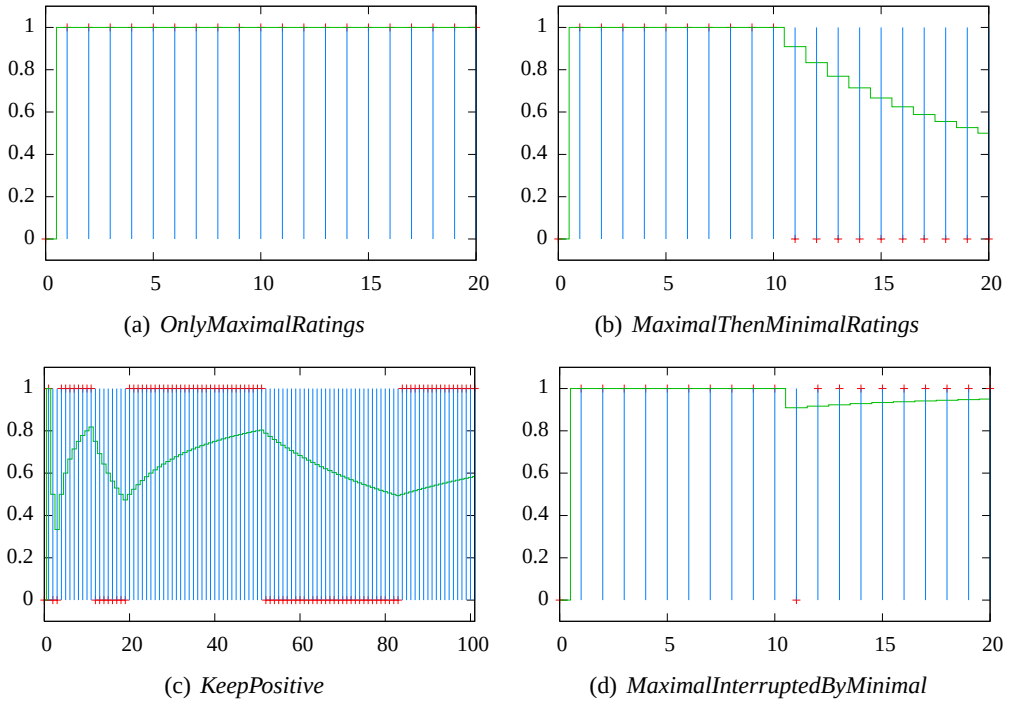


Abbildung 8.7.: Vertrauens- und Gewissheitsdynamik bei [Yu und Singh, 2002]

trauensberechnung angeht. Z.B. wird eine Berechnung der Gewissheit nicht konkretisiert und wenig zu der Interpretierbarkeit des Vertrauenswertes ausgesagt.

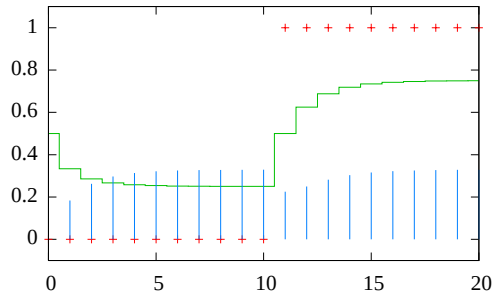
8.3.4. Jøsang [2002] (Beta Reputation System)

Skalierung in Abhängigkeit zu dem Alterungsfaktor

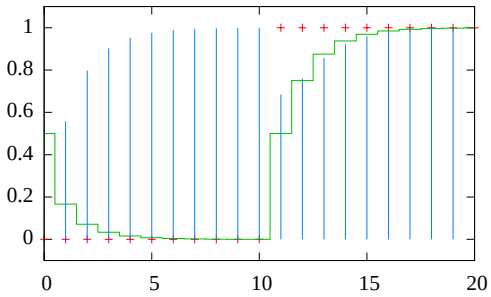
Wenn die Gewissheit und das Vertrauen entsprechend den vorgeschlagenen Skalierungen der Metriken (siehe Abschnitt 7.2.5) berechnet werden, ergibt sich eine Änderung an der Dynamik, die in Abbildung 8.8 zu erkennen ist.

MaximalInterrupted-ByMinimal, *MinimalThen-MaximalRatings*

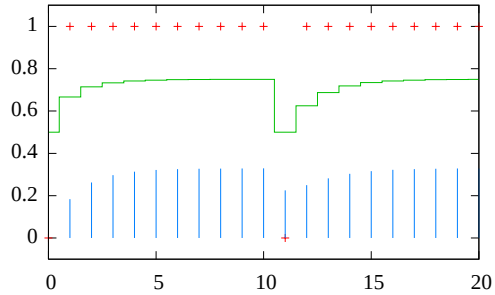
In Abbildungen 8.8(c) und 8.8(a) wird die normale Variante aus dem Beta Reputation System verwendet. In Abbildungen 8.8(b) und 8.8(d) wird die Erweiterung mit der Skalierung verwendet. Im Vergleich der beiden Varianten wird deutlich, wie mit der Änderung nun die kompletten Vertrauens- und Gewissheitsmetriken ausgeschöpft werden können. Dadurch tragen neue Bewertungen jedoch deutlich mehr zu einer Vertrauensänderung bei, was die gesamte Vertrauensdynamik etwas verschärft. Um diesen Effekt wieder entgegenzuwirken kann ein geringerer Alterungsfaktor gewählt werden.



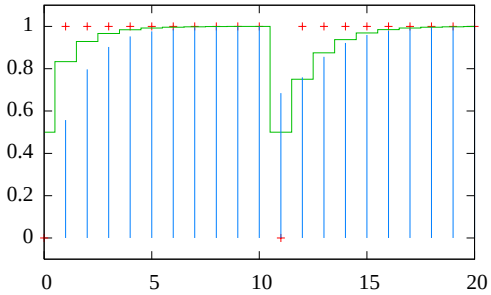
(a) Ursprüngliche Berechnung (Testszenario *MinimalThenMaximal*)



(b) Mit der Skalierungserweiterung (Testszenario *MinimalThenMaximal*)

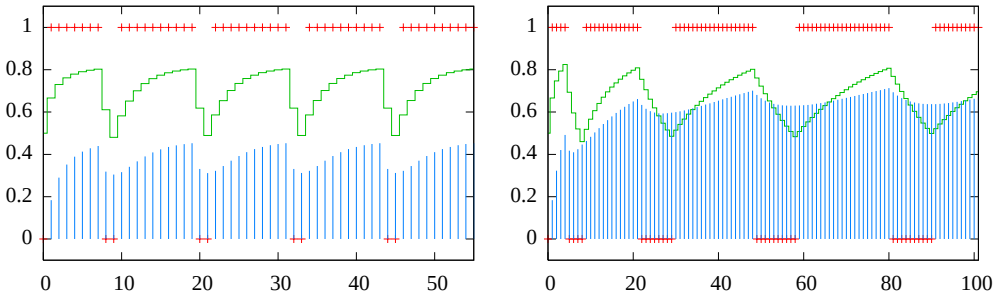


(c) Ursprüngliche Berechnung (Testszenario *MaximalInterruptedByMinimal*)



(d) Mit der Skalierungserweiterung (Testszenario *MaximalInterruptedByMinimal*)

Abbildung 8.8.: Vertrauens- und Gewissheitsdynamik bei dem Beta Reputation System mit Gewichtung $g = 1.0$ und Alterungsfaktor $\alpha = 0.5$



(a) KeepPositive mit Alterungsfaktor $\alpha = 0.3$ (b) KeepPositive mit Alterungsfaktor $\alpha = 0.05$

Abbildung 8.9.: Vertrauens- und Gewissheitsdynamik bei dem Beta Reputation System im Testszenario *KeepPositive*

Analyse

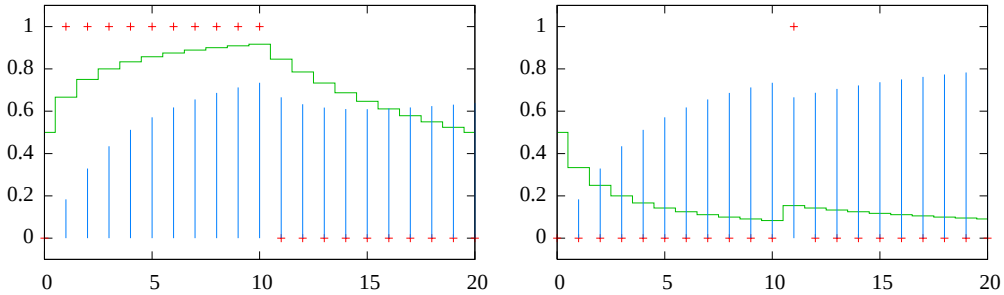
Im Allgemeinen spiegelt die Vertrauensdynamik hier sehr realistisch den Trend der Bewertungen wieder. Mit dem Einbringen eines Alterungsfaktors wird auch schnell auf eine Umschwankung des Bewertungstrends reagiert. Schon bei einen Faktor von 0.3 zeigt sich, dass eine zuvor aufgebaute gute Vertrauensbeziehung (mit einen Vertrauenswert von über 0.8) schon nach zwei minimalen Bewertungen wieder in den Misstrauensbereich gezogen wird (Abbildung 8.9(a)). Bei geringeren Alterungsfaktoren (z.B. 0.05 in Abbildung 8.9(b)) hingegen ist die Reaktion etwas langsamer, jedoch schneller als bei dem UniTEC-Algorithmus mit gleichem Alterungsfaktor. *KeepPositive*

Bei ausgeschaltenem Alterungsfaktor spielt wie erwartet die Reihenfolge der Bewertungen für den endgültigem Vertrauenswert keine Rolle mehr (da somit alle Bewertungen gleichwertig behandelt werden). Dieses wird im Vergleich der Endwerte bei den vier Szenarien in Abbildung 8.18 (auf Seite 87) deutlich.

Die Gewissheitsmetrik, die auf der Berechnung der Standardabweichung der Beta Verteilung aufbaut (wie in [Jøsang und McAnally, 2004] beschrieben), liefert Ergebnisse, die den Erwartungen entsprechen. Trendänderungen (Abbildung 8.10(a)) oder -unterbrechungen (Abbildung 8.10(b)) werden mit einem Gewissheitsverlust geahndet. Langfristig steigt die Gewissheit, wie zu erwarten, mit zunehmenden Anzahl von Bewertungen.

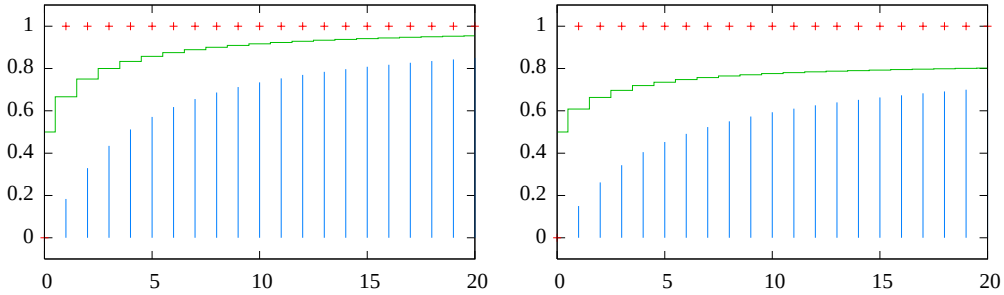
Das Gewichten einzelner Bewertungen zeigt ebenfalls das erwartete Verhalten: Eine schwach gewichtete Bewertung beeinflusst das Vertrauensverhältnis in geringerem Maße. Auch die Gewissheit wird bei geringerem Gewicht schwächer beeinflusst. In den Testergebnisse in Abbildung 8.11 wird dieses deutlich. *OnlyMaximal*

Fazit Hier ist eine gute Unterstützung des Alterungsfaktors und einer Gewichtung vorhanden, die sowohl den Vertrauenswert, wie auch dessen Gewissheit beeinflusst. Das Vertrauensmodell basiert auf mathematisch fundierten Modellen, und setzt diese sinnvoll in eine praktische Erfahrung um. Verbesserungswürdig wäre noch eine Einstellungsmöglichkeit wie bei Bin Yu [2000], wo negative Erfahrungen stärker als positive Erfahrungen gewichtet werden. Problematisch er-

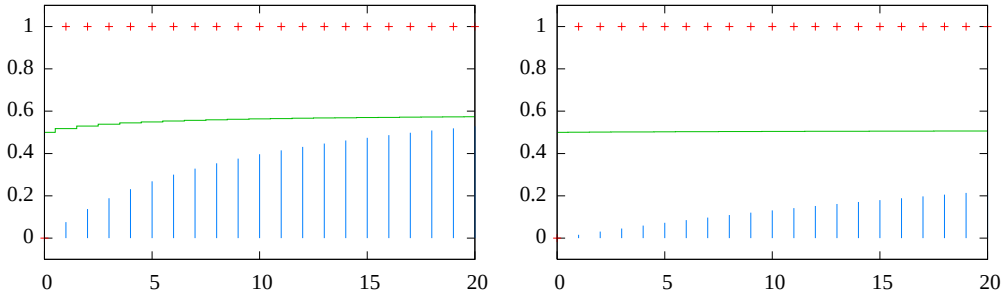


(a) *MaximalThenMinimalRatings* (b) *MinimalInterruptedByMaximal*

Abbildung 8.10.: Vertrauens- und Gewissheitsdynamik bei dem Beta Reputation System ohne Alterungseffekt



(a) Gewichtung von 1.0 (b) Gewichtung von 0.8



(c) Gewichtung von 0.3 (d) Gewichtung von 0.05

Abbildung 8.11.: Vertrauens- und Gewissheitsdynamik bei dem Beta Reputation System mit unterschiedlichen Gewichtungen im Testszenario *OnlyMaximalRatings*

scheint zudem noch die Annahme, dass der initiale Vertrauenswert im neutralen Vertrauensverhältnis angesiedelt ist (0.5), was im UniTEC-System nach einem Vertrauensverlust durch einem Pseudonymwechsel ausgenutzt werden könnte.

8.3.5. Sabater [2003] (ReGreT)

Durch eine durchgängige Nutzung einer Alterungsfunktion spielt die Reihenfolge der Bewertungen stets eine Rolle bei dem Endergebnis der Bewertungen. Dies zeigt sich z.B. bei den Testszenarien *SpecificRatings* (Abbildungen 8.12(a) und 8.12(b)).

SpecificRatings

Bei wachsender Bewertungshistorie reagiert der Algorithmus langsamer auf eine Änderung des Trends der Bewertungen. In Abbildung 8.12(c) wird dieses deutlich. Ursache dafür ist die ausgewählte Alterungsfunktion, die in [Sabater, 2003] als ein einfaches Beispiel dafür erwähnt wird und die bei der Implementierung verwendet wurde. Die alterungsbedingte Gewichtung der Bewertungen ist hier abhängig von der Bewertungshistorienlänge.

KeepPositive

Eine Gewichtung der Bewertungen wird in [Sabater, 2003] nicht beschrieben, konnte jedoch hinzugefügt werden. Sie erweist sich in den Testszenarien als sinnvolle Ergänzung. Eine kleinere Gewichtung der Bewertungen bewirkt wie erwartet eine Abschwächung der Vertrauensdynamik. Ein Beispiel hierfür ist in Abbildungen 8.12(d) und 8.12(e) zu sehen.

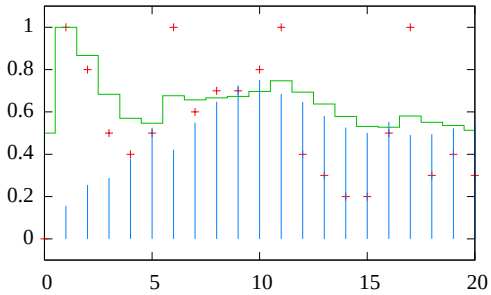
MaximalThen-MinimalRatings

Die Berechnung der Gewissheit basiert auf zwei Faktoren: Die Anzahl der Begegnungen (*number of outcomes factor*) und die Abweichung der Bewertungen in der Bewertungshistorie von dem aktuellen Vertrauenswert (*outcome reputation deviation*). In den Testläufen wurde der *intimate level of interactions (itm)* auf zehn Begegnungen gesetzt. Bis zu der zehnten Begegnung ist daher ein sinusförmiger Anstieg der Gewissheit zu beobachten (siehe Abbildung 8.12(e)). Ab dann spielt nur noch die Abweichung der Bewertungswerte eine Rolle.

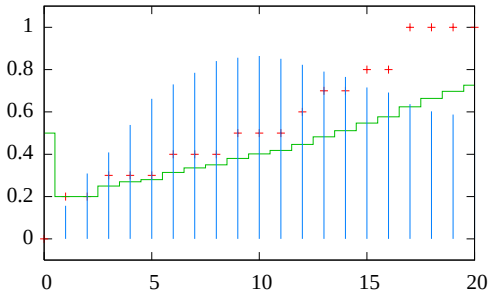
MaximalThen-MinimalRatings

Fazit Die Berechnung der Vertrauenswerte aus einem gewichteten Durchschnitt sämtlicher Bewertungen erweist sich als problematisch für eine schnelle Reaktion auf Trendänderungen: Je länger die Bewertungshistorie bereits ist, desto langsamer wird reagiert. Außerdem erreicht die beispielhafte Alterungsfunktion nicht den Aufwand von $O(1)$ pro Aktualisierungsvorgang. Hier könnte eine geeignete Alterungsfunktion gefunden werden, die eine inkrementelle Bearbeitung erlaubt (also vom Typ **Pseudoupdate**). Eine weitere mögliche Lösung wäre hier die Länge der Bewertungshistorien stets durch ein Historienfenster zu begrenzen (h). Dadurch könnte man den Aufwand des Algorithmus pro Aktualisierungsschritt auf $O(h)$ bringen.

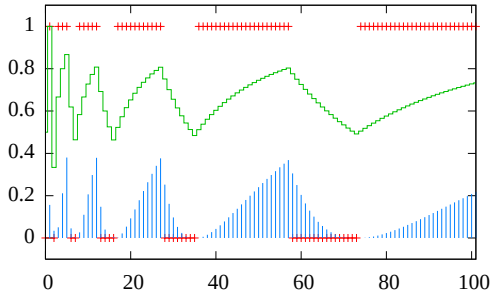
Die Gewissheit für die ersten itm Bewertungen zunächst inkrementell zu erhöhen ist für diese Art von Algorithmen, die stark von der Historienlänge abhängig sind, sinnvoll: Ein stetiges Verhalten (siehe Abschnitt 2.6) bei den ersten itm Begegnungen kann auch nur ein unglücklicher Zufall gewesen sein, der durch den *number of outcomes factor* ausgeglichen wird. Ab itm Begegnungen spielt aber wieder die Bewertungshistorienlänge einen negativen Einfluss auf die Gewissheit: Bei längerer Historie wird langsamer auf Trendänderungen reagiert.



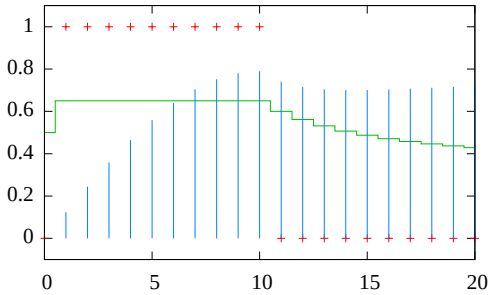
(a) Testszenario *SpecificRatingsNormal*



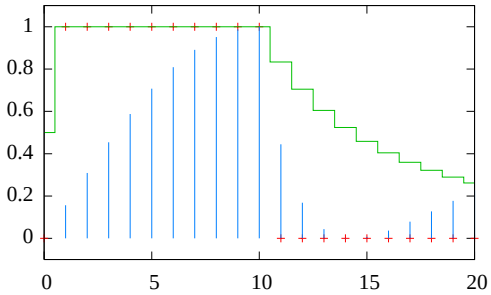
(b) Testszenario *SpecificRatingsOrderedAsc*



(c) Gewichtung 1.0 im Testszenario *KeepPositive*

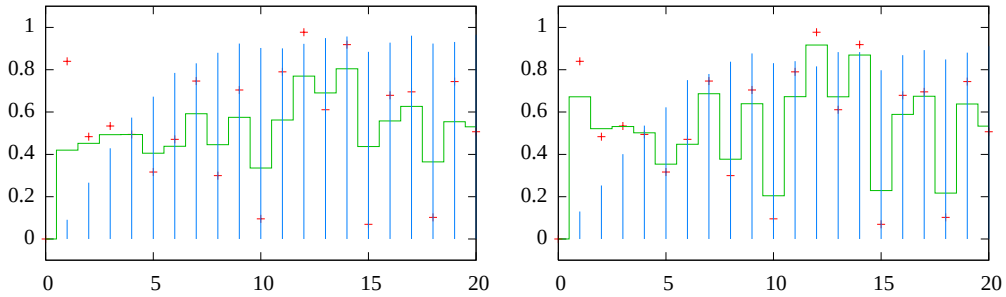


(d) Testszenario *MaximalThenMinimalRatings* mit Gewichtung 0.3



(e) Testszenario *MaximalThenMinimalRatings* mit Gewichtung 1.0

Abbildung 8.12.: Vertrauens- und Gewissheitsdynamik bei ReGreT



(a) Mit Alterungsfaktor $\alpha = 0.5$ (b) Mit Alterungsfaktor $\alpha = 0.8$

Abbildung 8.13.: Vertrauens- und Gewissheitsdynamik bei dem UniTEC-Algorithmus im Testszenario *RandomRatings*

8.3.6. Lik Mui [2003]

Für das Vertrauensmodell in [Mui, 2003] wurden keine getrennten Tests durchgeführt, da dieses Vertrauensmodell ein Spezialfall des *Beta Reputation Systems* ist. Das Besondere hier ist, dass es keine Alterung und Gewichtung der Bewertungen gibt. Außerdem sind nur binäre Bewertungen möglich.

Hier sind ebenfalls Überlegungen zu der groben diskreten Metrik nötig, wie schon bei Rahman [2000]. In diesem Fall jedoch ist lediglich die Bewertungsmetrik diskret: die Vertrauensmetrik ist wie auch bei dem Beta Reputation System skalar. Die Bewertungen werden lokal verarbeitet, somit kann der Bewertungsbereich auf die beiden Extremwerte der skalaren Metrik des generischen Vertrauensmodells eingeschränkt werden.

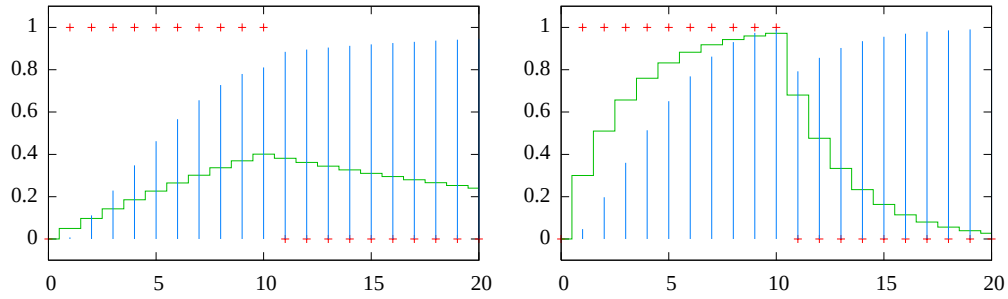
8.3.7. UniTEC [2003]

Der Algorithmus (geometrisches Lernen, UniTEC und Jonker) nähert sich bei Vergrössern des Alterungsfaktors einem Tit-for-Tat Algorithmus, wo die zuletzt bekannte Bewertung den aktuellen Vertrauenswert direkt ausmacht. Bei Alterungsfaktoren über 0.5 ist dieser Effekt schon deutlich zu erkennen (Abbildung 8.13). Bei einem zu geringen Alterungsfaktor verliert man jedoch stark an Reaktionsschnelligkeit: Es wird nicht entsprechend schnell auf einem Trendwechsel bei den Bewertungen reagiert. Dieses ist hauptsächlich bei einem Wechsel von positiven zu negativen Bewertungen problematisch, was bereits in [Kinateder und Rothermel, 2003] als verbesserungswürdig bei diesem Algorithmus erkannt wurde. Dies wird im Vergleich der Abbildungen 8.14(a) und 8.14(b) deutlich. Optimale Alterungsfaktoren liegen demzufolge in dem Bereich $0.3 < \alpha < 0.5$.

RandomRatings

MaximalThen-MinimalRatings

Die Gewissheit durch die Standardabweichung (Erweiterung zu UniTEC, siehe Abschnitt 7.2.7) berechnet hier einen Verlauf der Ähnlich mit den Verlauf bei ReGreT ist, da er auf ähnliche Berechnungsmechanismen beruht. Allerdings reagiert die Gewissheit bei einem Trendwechsel der Bewertungen bei einen sehr geringen Alterungsfaktor auch sehr langsam, teilweise auch gar



(a) Mit Alterungsfaktor $\alpha = 0.05$ (b) Mit Alterungsfaktor $\alpha = 0.3$

Abbildung 8.14.: Vertrauens- und Gewissheitsdynamik bei dem UniTEC-Algorithmus im Testszenario *MaximalThenMinimalRatings*

nicht (siehe Abbildung 8.14(a)). Dieser Effekt kann durch die Einschränkung auf den bereits erwähnten optimalen Bereich von $0.3 < \alpha < 0.5$ vermieden werden.

KeepPositive

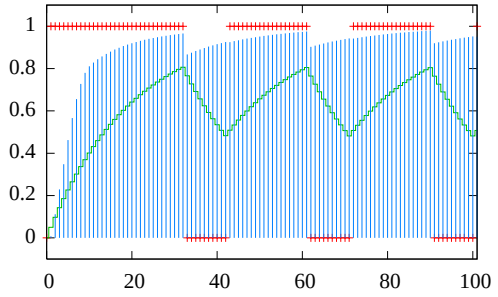
In Abbildung 8.15(a) ist zu erkennen, wie die Reaktion auf den Trendwechsel bei einem geringen Alterungsfaktor (0.05) zu viele Bewertungen erfordert (in diesem Fall sind zehn minimale Bewertungen notwendig, um den zuvor erreichten guten Vertrauenswert zunichte zu machen). Mit einem Alterungsfaktor von 0.3 (Abbildung 8.15(b)) und 0.5 (Abbildung 8.15(c)) ist die Reaktion besser (lediglich zwei bzw. eine negative Bewertungen notwendig als Reaktionszeit).

Die Gewichtung der Bewertungen schafft den gewünschten Effekt. Alle Arten der Vertrauensdynamik werden hiermit lediglich in ihrer Schärfe abgeschwächt.

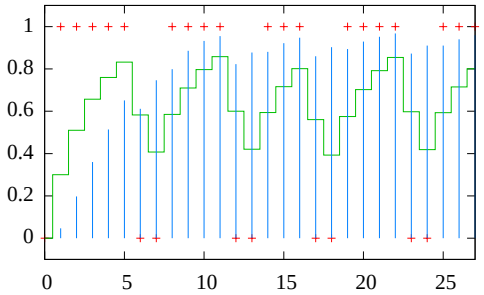
Abnutzungsfaktor

Der Abnutzungsfaktor wurde im UniTEC-Algorithmus testweise eingebaut. Da es sich hier um einen **Updatealgorithmus** handelt, lässt sich die Abnutzung direkt auf den Vertrauenswert auswirken. Die Gewissheit wurde in dieser Umsetzung nicht betrachtet. Als Testszenario dient eine Bewertungshistorie, wo nur zu bestimmten Zeitpunkten eine Bewertung abgegeben wird. In den restlichen Zeitpunkten wird lediglich der aktuelle Vertrauenswert abgefragt. In Abbildung 8.16 zeigt sich die Auswirkung zweier unterschiedlicher Abnutzungsfaktoren. Die Zeiteinheit der Abnutzung ist stark abhängig von der Anwendung und der Dichte an Bewertungen. Bei einem zu großen Abnutzungsfaktor mit zu geringem Alterungsfaktor zeigt sich, dass sich das Vertrauen nicht mehr aufbauen lässt (siehe Abbildung 8.17). In solchen Fällen kann lediglich eine höhere Bewertungsdichte hierauf noch Einfluss nehmen. Deswegen ist die Einstellung des Abnutzungsfaktors bei unbekannter Bewertungsdichte schwierig.

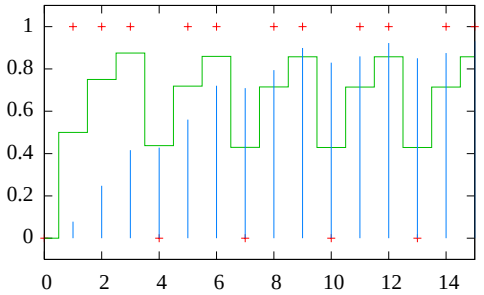
Fazit Es kann in diesem Vertrauensmodell nicht die komplette Bandbreite des Alterungsfaktor genutzt werden. Nur ein bestimmter Bereich ergibt für die Vertrauensanwendung tatsächlich einen Sinn. Auch ein komplettes Ausschalten des Alterungsfaktors ist nicht möglich. Die Gewichtung der Bewertungen erfüllt die gewünschte Zielsetzung.



(a) Mit Alterungsfaktor $\alpha = 0.05$

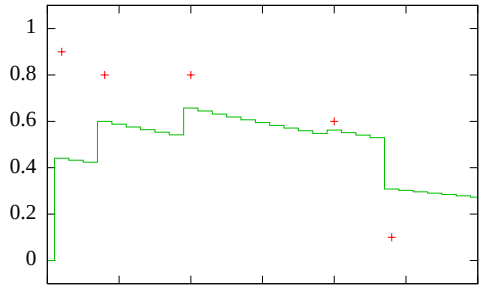


(b) Mit Alterungsfaktor $\alpha = 0.3$

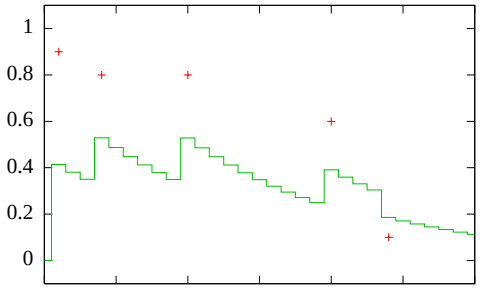


(c) Mit Alterungsfaktor $\alpha = 0.5$

Abbildung 8.15.: Vertrauens- und Gewissheitsdynamik bei dem UniTEC-Algorithmus im Testszenario *KeepPositive*

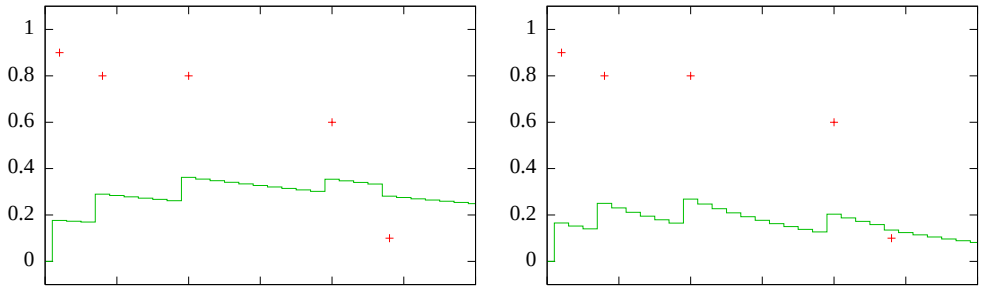


(a) Abnutzungsfaktor $\lambda = 0.1$



(b) Abnutzungsfaktor $\lambda = 0.4$

Abbildung 8.16.: Vertrauensdynamik bei UniTEC-Algorithmus mit einem Abnutzungsfaktor (bei Alterungsfaktor $\alpha = 0.5$)



(a) Abnutzungsfaktor $\lambda = 0.1$ (b) Abnutzungsfaktor $\lambda = 0.4$

Abbildung 8.17.: Vertrauensdynamik bei UniTEC-Algorithmus mit einem Abnutzungsfaktor (bei Alterungsfaktor $\alpha = 0.2$)

Bei einem Alterungsfaktor von $\alpha = 0.3$ zeigt der Algorithmus eine gute Nachempfindung des Bewertungstrends in praktisch allen getesteten Szenarien.

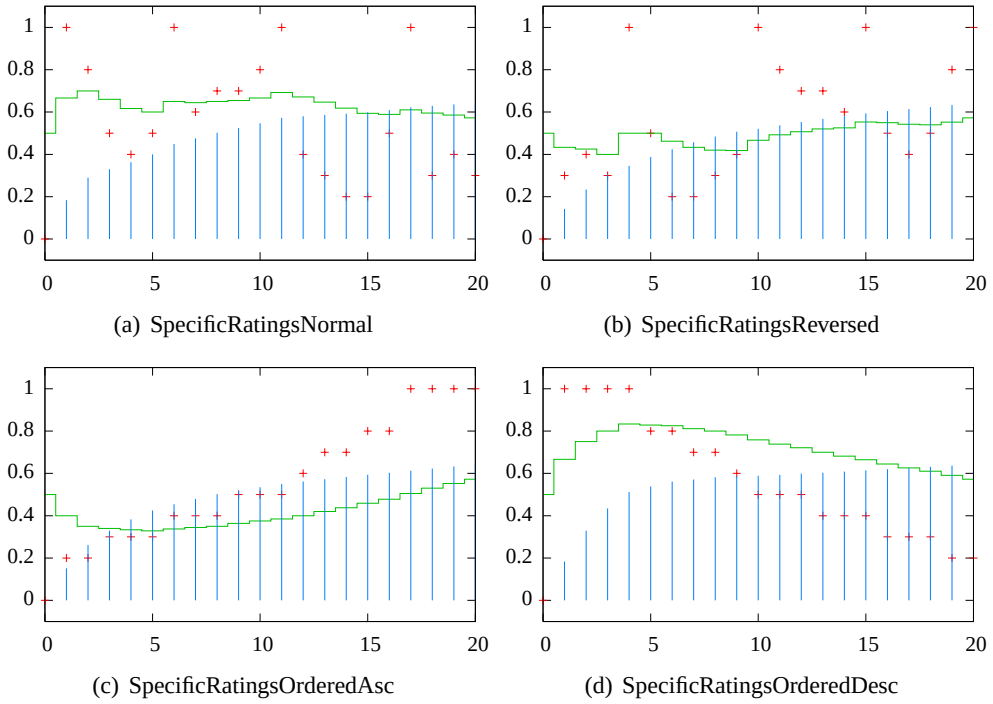
8.3.8. Allgemeine Ergebnisse

Die Reihenfolge der Bewertungen spielt in folgenden Situationen auf den letztendlichen Vertrauenswert keine Rolle:

- Bei Algorithmen, die keinen Alterungsfaktor vorsehen.
- Bei dem Beta Reputation System mit Alterungsfaktor 0.0 (d.h. kein Alterungseffekt). In Abbildung 8.18 ist zu erkennen, dass bei den vier unterschiedlichen Reihenfolgen der selben Menge an Bewertungen der Endvertrauenswert der selbe ist.

Bei dem UniTEC-Algorithmus bewirkt der Alterungsfaktor 0.0, dass überhaupt keine Bewertung berücksichtigt werden, und sich daher auch kein Vertrauen aufbaut. Die Alterung lässt sich also nicht ausschalten, und daher spielt die Reihenfolge der Bewertungen stets eine Rolle.

In Abbildungen 8.19 und 8.20 wird ein Vergleich der betrachteten Vertrauensalgorithmen im Testszenario *MaximalThenMinimalRatings* gezeigt. Das Vertrauen wird zunächst mit maximalen Bewertungen aufgebaut, anschliessend mit minimalen Bewertungen wieder bestraft. Der Algorithmus Bin Yu [2000] bei einer stärkeren Gewichtung positiver Erfahrungen (was in [Yu und Singh, 2000] auch nicht empfohlen wird) reagiert überhaupt nicht auf den Trendwechsel. Extrem langsam ist die Reaktion bei Bin Yu [2000] (bei Gleichgewichtung positiver und negativer Bewertungen), Rahman, Bin Yu [2002] und bei dem Beta Reputation System ohne Alterung oder mit einer sehr geringen Alterung ($\alpha = 0.05$) zu beobachten. Es sind bei diesen Kandidaten jeweils mehr als sieben minimale Bewertungen notwendig, um das Vertrauen in den Misstrauensbereich zu bringen. Bei ReGreT ist ein ebenfalls ein relativ langsamer Abbau des Vertrauens deutlich, erreicht jedoch bereits nach fünf minimalen Bewertungen den Misstrauensbereich. Bei



(a) SpecificRatingsNormal (b) SpecificRatingsReversed

(c) SpecificRatingsOrderedAsc (d) SpecificRatingsOrderedDesc

Abbildung 8.18.: Vertrauens- und Gewissheitsverlauf des Beta Reputation System bei den *SpecificRatings* Testszenarien mit Alterungsfaktor $\alpha = 0.0$

den anderen Kandidaten ist innerhalb von einer bis vier minimalen Bewertungen der Misstrauensbereich erreicht.

In Abbildungen 8.21 und 8.22 ist ein realistisches Testszenario mit willkürlichen Bewertungen dargestellt (Testszenario *SpecificRatingsNormal*). In der ersten Hälfte des Szenarios sind deutlich mehr positive Bewertungen zu erkennen, während in der zweiten Hälfte einige negative Bewertungen das Vertrauen wieder absenken lassen sollen. Rein intuitiv (aus Betrachtung der Bewertungsfolge) ist ein Endvertrauenswert im Bereich $0.3 < T < 0.5$ zu erwarten. Es ist zu erkennen, wie die viele Testkandidaten dieses Szenario gut überstehen und dabei einen Endwert in dem erwartetem Bereich erreichen. Einige Ausnahmen bleiben jedoch zu Erwähnen: Bin Yu [2000] mit starker positive Gewichtung (*pos*) und mit Gleichgewichtung positiver und negativer Bewertungen schwankt relativ schnell zu einem absoluten Vertrauen und lässt sich kaum wieder zurückbringen. Bin Yu [2000] mit einer stärkeren Gewichtung negativer Bewertungen (*neg*) bestraft die zweite Hälfte der Bewertungen deutlich krasser als alle anderen Kandidaten: Das Vertrauensverhältnis endet knapp über dem absoluten Misstrauen. UniTEC mit einem zu geringen Alterungsfaktor schafft es kaum eine Vertrauensdynamik zu entwickeln, was bereits in den vorherigen Erkenntnissen erwähnt wurde. Rahman schafft es trotz der diskreten und sehr ungenauen Metrik eine realistische Vertrauensdynamik zu entwickeln. Alle Beta Reputation System Varianten erstellen ähnlich realistische Vertrauensdynamiken.

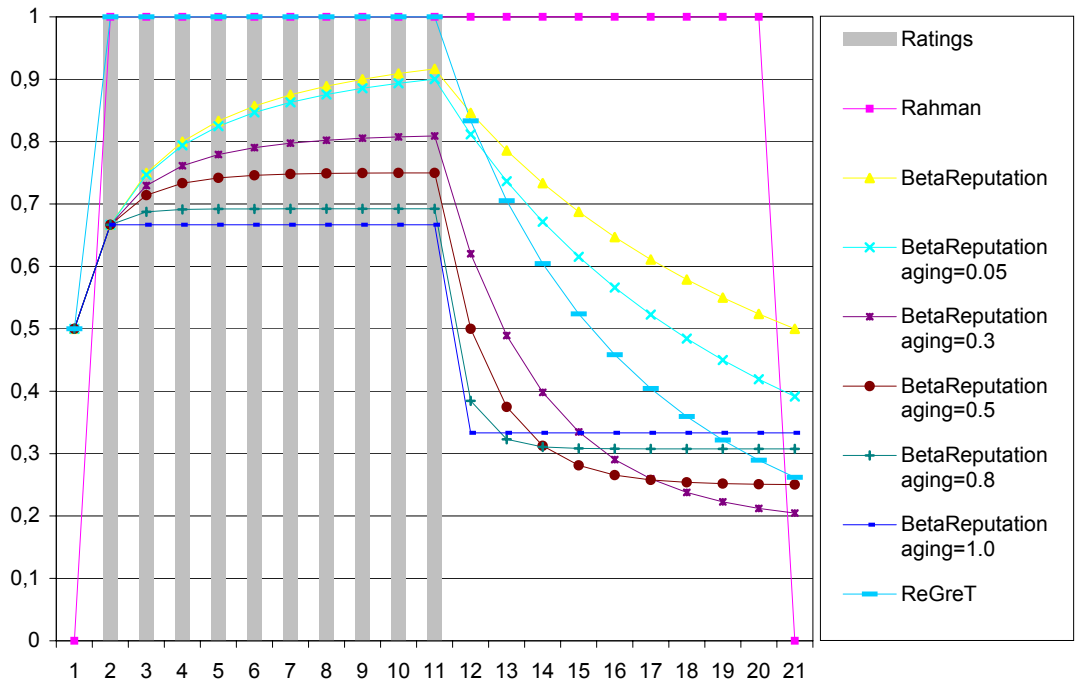


Abbildung 8.19.: Vertrauensverlauf bei Testszenario *MaximalThenMinimalRatings*, Teil 1

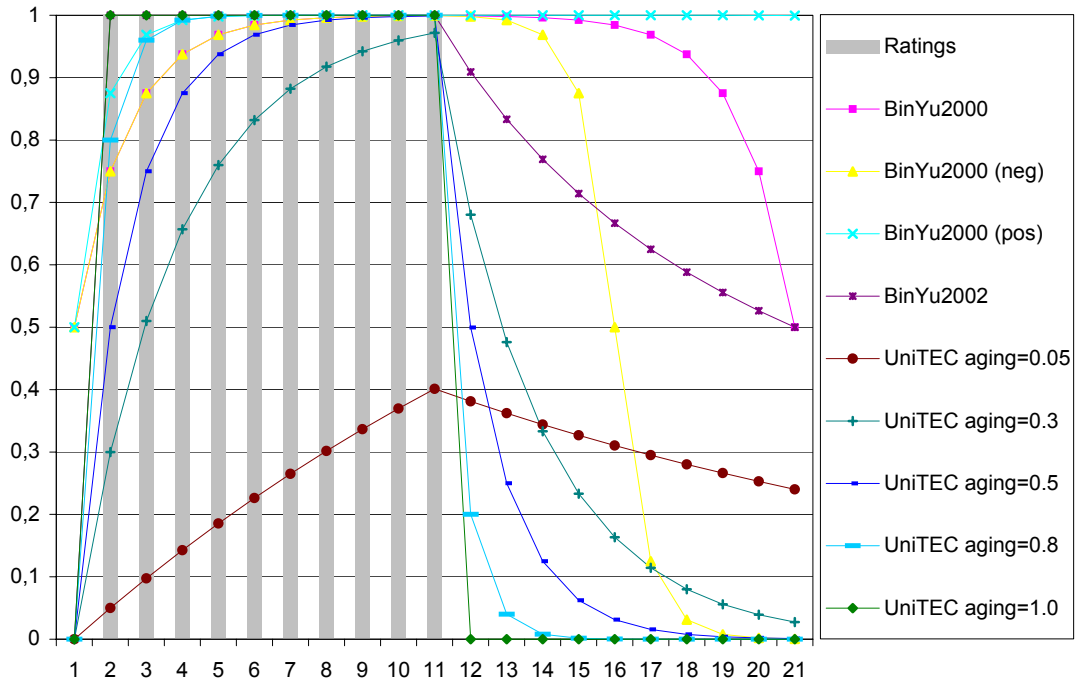


Abbildung 8.20.: Vertrauensverlauf bei Testszenario *MaximalThenMinimalRatings*, Teil 2

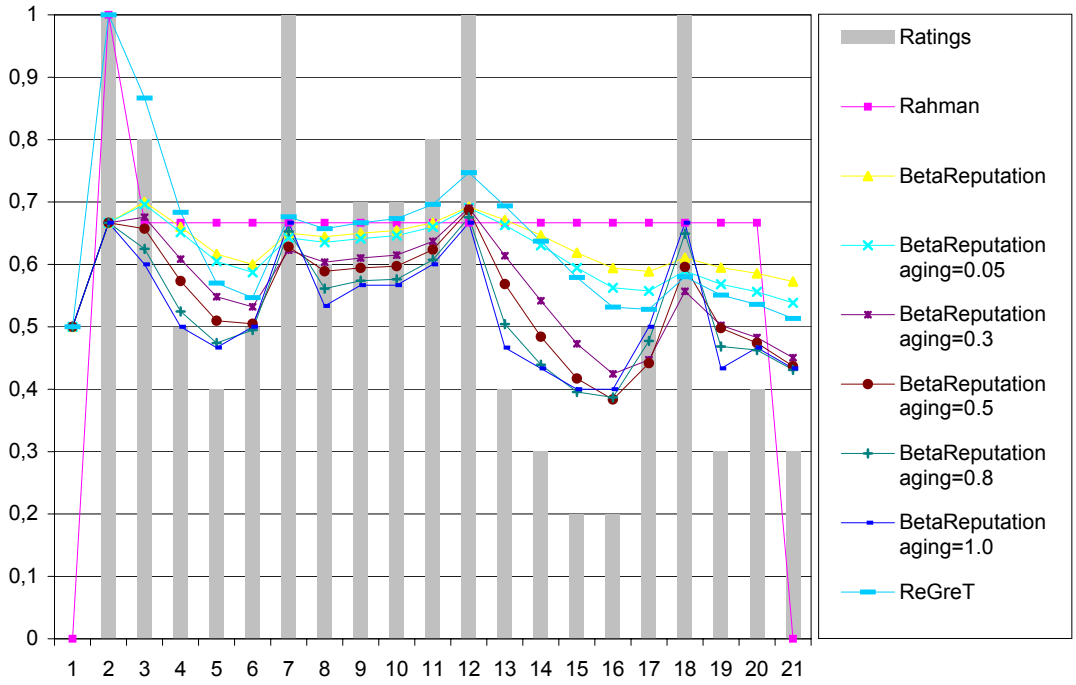


Abbildung 8.21.: Vertrauensverlauf bei Testszenario *SpecificRatingsNormal*, Teil 1

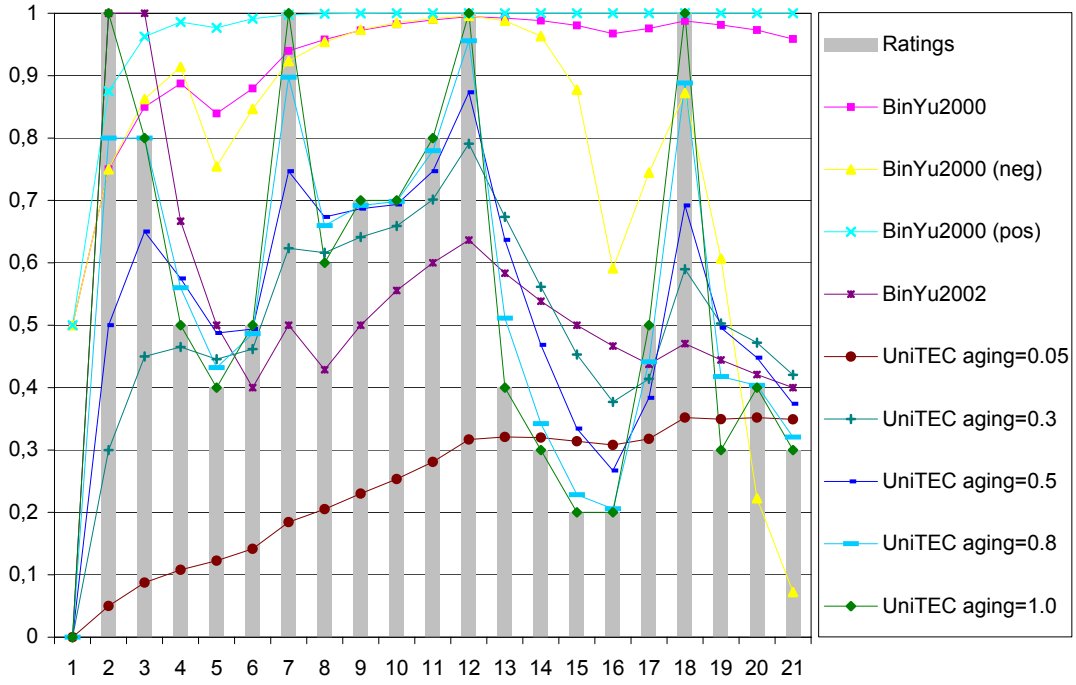


Abbildung 8.22.: Vertrauensverlauf bei Testszenario *SpecificRatingsNormal*, Teil 2

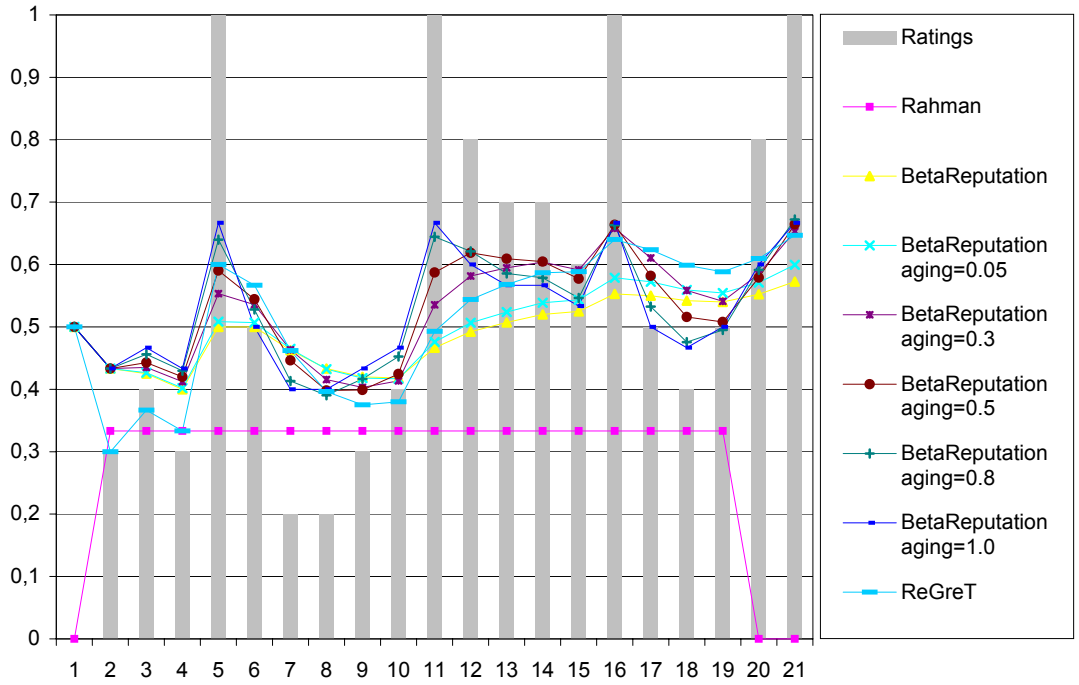


Abbildung 8.23.: Vertrauensverlauf bei Testszenario *SpecificRatingsReversed*, Teil 1

Nun wurde die gleiche Bewertungsfolge umgedreht, so dass in der ersten Hälfte deutlich mehr negative Bewertungen abgegeben wurden als in der zweiten Hälfte (Testszenario *SpecificRatingsReversed* in Abbildungen 8.23 und 8.24). Als Spiegelung zu dem vorherigen Szenario ist hier rein intuitiv ein Endvertrauenswert im Bereich $0.5 < T < 0.7$ zu erwarten. Auch hier sind ähnliche Beobachtungen wie zu dem vorherigem Fall zu machen. Bin Yu [2000] (*pos*) erreicht relativ schnell das blinde Vertrauen, während Bin Yu [2000] (*neg*) das Vertrauen nahe am absoluten Misstrauen belässt. Bin Yu [2002] und UniTEC mit schwachem Alterungsfaktor (0.05) bleiben im Misstrauensbereich. Bei Rahman erreicht das Vertrauensverhältnis durch die Konstelation der Bewertungen das positive Vertrauen ebenfalls nicht. Die Beta Reputation System Varianten erreichen Endwerte, die unseren intuitive Erwartungen entsprechen, während weitere UniTEC Varianten und Bin Yu [2000] eher zu optimistisch das Endvertrauensverhältnis einschätzen.

8.4. Zusammenfassung der Ergebnisse

Aus jedem der Vertrauensmodelle konnten interessante Erkenntnisse zu der Modellierung von Vertrauen auf Rechnersysteme gewonnen werden.

Die Funktion des Beta Reputation System (und damit auch Lik Mui [2003]) erwies sich im Allgemeinen als momentan sinnvollste Variante. Die statistische Grundlagen (Bayesche Wahrscheilichkeit) geben dem Modell eine fundierte mathematische Basis. Die Erkenntnisse aus den

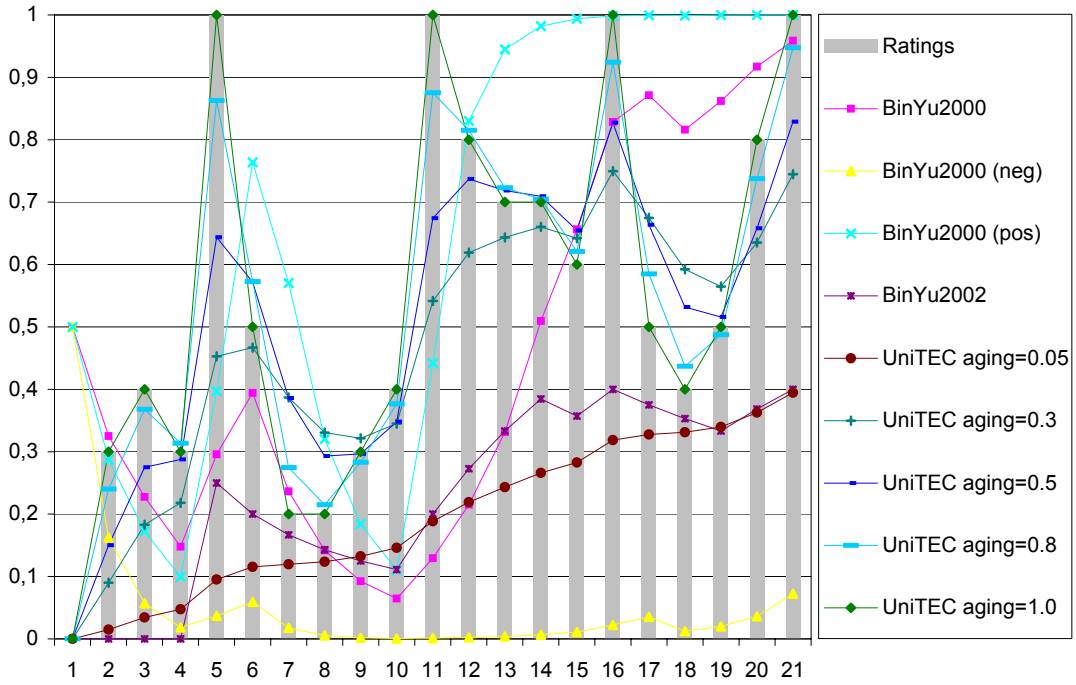


Abbildung 8.24.: Vertrauensverlauf bei Testszenario *SpecificRatingsReversed*, Teil 2

Testszenarien zeigen dessen Vorteile im praktischem Einsatz. Allerdings sollten die vorgeschlagenen Änderungen zu der Skalierung betrachtet werden, damit die Vertrauens- und Gewissheitswerte nicht künstlich unabsichtlich begrenzt werden. Die Idee, eine Unterscheidung zwischen negativen und positiven Erfahrungen, wie bei Bin Yu [2000] realisiert, könnte ebenfalls mit einfliessen. Damit könnte der menschliche Aspekt eingebaut werden, bei dem schlechte Erfahrungen stärker Vertrauensentscheidungen beeinflussen als positive. Die Initialisierung des Vertrauenswertes sollte auf jeden Fall einen potentiellen Missbrauch verhindern, was bei UniTEC explizit erwähnt und berücksichtigt wird.

Die Berechnung der Gewissheit im Beta Reputation System zeigt ebenfalls gute Ergebnisse. Zukünftige Arbeiten könnten die Gewissheit zusätzlich mit der Bewertungshistorienlänge oder anderen Parametern koppeln, so dass der Benutzer (oder die spezifische Anwendung, wo die Vertrauensberechnung verwendet wird) hierauf mehr Einfluss nehmen kann, z.B. eine Mindestanzahl von Bewertungen, die notwendig sind, um überhaupt ein Mindesgewissheit erreichen zu können. Die Lösungen bei ReGreT und die Vorschläge zur Erweiterung des UniTEC-Algorithmus bringen dazu die notwendige Anreize.

Grössere Schwierigkeiten bei der Vergleichbarkeit bereiten Vertrauensmodelle mit groben diskreten Metriken. Es muss bei der Umrechnung die Bedeutung der einzelnen diskreten Werte berücksichtigt werden, um Vergleiche ziehen zu können. Viele Werte der skalaren Metrik lassen sich nicht sinnvoll in der diskreten Metrik darstellen.

Wenn richtig eingesetzt, eröffnet der neu eingeführte Abnutzungsfaktor neue Möglichkeiten.

Er ist jedoch sehr anwendungsspezifisch und kann nur durch Kenntniss der potentiellen Bewertungsdichte sinnvoll eingesetzt werden.

9. Fazit und Ausblick

9.1. Fazit

Nach einer Einführung in die Vertrauensthematik wurden wichtige Bestandteile von Vertrauensmodellen beschrieben. Daraus wurde ein generisches Vertrauensmodell erarbeitet, welches die Aspekte von Vertrauensmodellen vereinheitlichen konnte. Durch den Einsatz des generischem Vertrauensmodells in UniTEC konnte eine Auswahl an Vertrauensmodellen aus der Literatur integriert werden. Im generischen Vertrauensmodell wird eine einheitliche Metrik definiert, die erstmalig eine direkte Vergleichbarkeit der unterschiedlichen Vertrauensaktualisierungsalgorithmen zulässt. Zusätzlich können dadurch ebenfalls erstmalig unterschiedliche Vertrauensmodelle in einem verteilten System miteinander kommunizieren: Jeder Benutzer kann dadurch ein anderes *lokales Vertrauensmodell* verwenden, und trotzdem erfolgreich im UniTEC-Netz teilnehmen.

Die Architektur der Vertrauenskomponente in UniTEC ermöglicht es nun mit minimalsten Aufwand neue Vertrauensmodelle umzusetzen und die entsprechenden Algorithmen somit zu testen. Auch lokal wird der Vorteil der generischen Metriken und der Vertrauensschnittstelle deutlich: Sämtliche UniTEC-Komponenten können auf eine einheitliche Weise auf Vertrauensinformationen zugreifen. Das generische Vertrauensmodell schirmt die Komplexität des unterliegenden lokalen Vertrauensmodells ab und bietet eine einheitliche und einfache Schnittstelle zu Vertrauensinformationen.

Durch den vergleichenden Untersuchungen der betrachteten Vertrauensmodelle und Vertrauensalgorithmen konnten wichtige Merkmake, Vorteile und auch Nachteile erkannt werden. Zu einzelnen Vertrauensmodellen und -algorithmen konnten einige Verbesserungs- und Erweiterungsmöglichkeiten erkannt und umgesetzt werden, die letztendlich auch evaluiert wurden.

9.2. Ausblick

Mit dem generischem Vertrauensmodell ist ein weiterer wichtiger Schritt zu einem besseren Verständnis von Vertrauensmodellierung geschaffen. Es bleibt weiterhin interessant die Erkenntnisse aus der Analyse zu nutzen, um daraus existierende Vertrauensmodelle weiter zu verfeinern oder komplett neue Modelle und Berechnungsmethoden zu erstellen, mit dem Ziel die Modellierung „menschlicher“ zu gestalten. Dazu bietet UniTEC bereits ein nützliches und sehr mächtiges Framework.

Die Berechnung der Gewissheit einer Vertrauensaussage ist ein Aspekt, der noch genauer betrachtet werden könnte. Ebenso wie das Vertrauen ist die Gewissheit sehr subjektiv und kann

von Person und Person unterschiedliche Bedeutungen und Ausprägungen haben.

Weiterhin bietet die Interpretation der Vertrauens- und der Gewissheitsinformationen interessantes Forschungspotential. Wie können aus den Vertrauensinformationen Vertrauensentscheidungen automatisch erzeugt werden? Welche Bedingungen sind in welchen Situationen dafür zu berücksichtigen? Wie können bestimmte Vertrauensketten interpretiert werden, wie sieht es mit der Transitivität von Vertrauen aus? Wie spielt die Wichtigkeit (*utility*) von Auswirkungen der Vertrauensentscheidungen eine Rolle bei Vertrauensberechnungen und die daraus zu folgernde Vertrauensentscheidungen?

Viele der betrachteten Vertrauensmodelle basieren auf Informationen, die durch Anfragen aus dem Netz erhalten werden. Es könnte eine Vereinheitlichung der gängigen Kommunikationsvarianten zur Vertrauenserhebung angestrebt werden, damit jedes lokale Vertrauensmodell solche Anfragen von allen beteiligten Vertrauensmodelle verarbeiten und beantworten kann.

Unterschiedliche Vertrauensanwendungen sollten analysiert werden, und wie diese mit dem generischen Vertrauensmodell zusammenspielen. In vielen der in der Literatur vorgestellten Vertrauensmodelle wird jeweils eine bestimmte Anwendung betrachtet: Unterschiedliche Anwendungen erfordern eventuell unterschiedliche Parameter oder unterschiedliche Berechnungsmechanismen. Welche Aspekte sind in welchen Anwendungen relevant? Reichen unterschiedliche Kontexte für unterschiedliche Anwendungskreise?

Auch bei der Kontextbetrachtung bleibt Raum für weitere Arbeiten. Das in UniTEC verwendete Kategorienmodell könnte erweitert und verfeinert werden, damit mehr Informationen zu den Beziehungen zwischen verwandten Kategorien zur Verfügung stehen. Diese Informationen können sich unterschiedlich auf die Vertrauensberechnung (oder -entscheidungen) auswirken. Wenn z.B. erkannt werden könnte, ob eine Kategorie generisch oder sehr spezialisiert ist, kann diese Information bei einem Vergleich unterschiedlicher Bewertungen durchaus relevant sein, gerade wenn die betrachtete Person in diesem Kontext ein wechselhaftiges Verhalten aufweist.

Das generische Vertrauensmodell im UniTEC-System bietet eine einfache Möglichkeit mit neuen Ideen in dem Bereich von Vertrauen und Reputation zu experimentieren und diese auszutesten.

A. Java Dokumentation

A.1. Kategorienkomponente, Paket `unitec.categories`

A.1.1. Basiselemente

Basisobjekte in diesem Paket sind einzelne Kategorien. Jede Kategorie wird durch eine Instanz der Klasse `Category` beschrieben. Ein semantischer Abstand zwischen zwei Kategorien wird durch die Klasse `CategoryDistance` beschrieben, die auf zwei Kategorien verweist (die Quelle und Ziel) und den entsprechenden Abstand als skalarer Wert im Bereich $[0, 1]$ angibt. Instanzen dieser beiden Klassen können nur über den `CategoryAccessor` erhalten werden. Diese zwei Basisklassen werden in der Abbildung A.1 dargestellt.

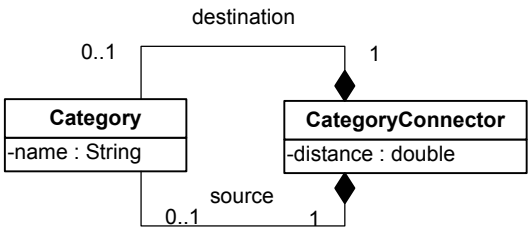


Abbildung A.1.: Basisklassen in `com.hp.unitec.category`

com.hp.unitec.category

public class **Category**

Methoden

String	**getName** ()	Gibt den Namen dieser Kategorie zurück.
void	**setName** (String name)	Setzt den Namen dieser Kategorie.
double	**getAgingFactor** ()	Gibt den Alterungsfaktor im Bereich $[0..1]$, der zu dieser Kategorie gehört zurück.
void	**setAgingFactor** (double agingFactor)	Setzt den Alterungsfaktor im Bereich $[0..1]$, der zu dieser Kategorie gehört.

A.1.2. Manager

Die Schnittstelle `CategoryAccessor` bietet den lesenden Zugang zu den Kategorien an. Eine zweite Schnittstelle - `CategoryManager` - bietet Methoden zur Verwaltung der Kategorien an (Kategorien Hinzufügen, Verknüpfen, Einlesen, Speichern, usw.).

Beide Schnittstellen werden zunächst auf Basis des Pakets Jung (vgl. White u. a. [2004]) implementiert. Diese Implementierungen werden im Paket `com.hp.unitec.categories.jung` untergebracht.

Der Klient der eine der beiden Schnittstellen verwenden will, kann sich ein Objekt, welches die entsprechende Schnittstelle implementiert über die Fabrik `CategoryManagerFactory` holen. Dieser kennt die möglichen Implementierungen, und erzeugt bei Bedarf eine Instanz davon.

com.hp.unitec.category

public interface **CategoryAccessor**

Methoden

Set	**getCategories** ()	Gibt die Menge aller Kategorien (alle Instanzen von `Category`) zurück.
Category	**getCategory** (String name)	Sucht eine Kategorie nach ihren Namen und gibt die Instanz davon zurück.
Set	**getRelated** (Category category)	Gibt zu der gegebenen Kategorie eine Menge von `CategoryDistance` zurück, die alle semantischen Distanzen zwischen ihr und anderen Kategorie beschreibt, die eine positive semantische Distanz aufweisen.
Set	**getRelated** (Category category, double minDistance)	Gibt zu der gegebenen Kategorie eine Menge von `CategoryDistance` zurück, die alle semantischen Distanzen zwischen ihr und anderen Kategorie beschreibt, die eine semantische Distanz von mindestens `minDistance` aufweisen.

Zu Verwaltung der Kategorien dient die folgende Schnittstelle:

com.hp.unitec.category

public interface **CategoryManager**

Methoden

void **load** (Reader reader) Lädt die Kategorien über das genannte Reader-Objekt. Der Kategoriengraph ist in einer Abwandlung des GraphML-Formates (XML) gespeichert.

void **save** (Writer writer) Speichert den Kategorienbaum über das übergebene Writer-Objekt.

Category **addCategory** (String newName) throws CategoryNameCollisionException
Fügt eine Kategorie mit einem bestimmten Namen hinzu.

void **removeCategory** (String name) throws CategoryNotFoundException
Löscht eine Kategorie.

void **connectCategories** (Category source, Category target, double distance) throws CategoryNotFoundException
Verbindet zwei Kategorien semantisch mit dem Gewicht `distance`. Falls schon eine Verbindung existiert, wird diese durch die neue Distanz ersetzt.

boolean **areConnected** (Category source, Category target) throws CategoryNotFoundException
Zeigt, ob zwei Kategorien semantisch direkt verbunden sind.

Letztendlich folgt die Fabrik-Klasse zur Instanzierung der beiden Manager-Objekte für Kategorien:

com.hp.unitec.category

public class **CategoryManagerFactory**

Methoden

static CategoryAccessor
getCategoryAccessor ()
Gibt die Implementierung der CategoryAccessor-Schnittstelle zurück.

static CategoryManager
getCategoryManager ()
Gibt die Implementierung der CategoryManager-Schnittstelle zurück.

Die Klasse `JungCategoryManager` in `com.hp.unitec.category.jung` implementiert momentan als einzige Klasse beide Schnittstellen. Eine Instanz der gewünschten Schnittstelle wird

bei Bedarf von `CategoryManagerFactory` erzeugt und zur Verfügung gestellt.

Die Zusammenhänge der Klassen wird in der Abbildung A.2 deutlich.

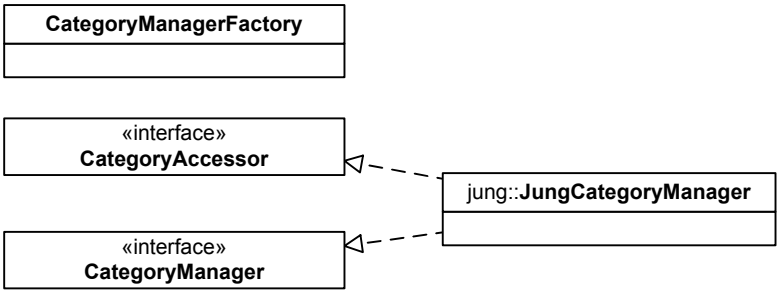


Abbildung A.2.: Kategorienverwaltung in `com.hp.unitec.category`

A.2. Vertrauenskomponente, Paket `unitec.trust`

Die Vertrauenskomponente ist der Kern der Vertrauensberechnungen. In ihr werden die Vertrauensverhältnisse zwischen dem lokalen Benutzer und fremden Pseudonymen berechnet, gespeichert und verwaltet. Der Benutzer hat auf diese Berechnungen Einfluss, indem er Empfehlungen fremder Pseudonyme über die Bewertungskomponente (siehe Abschnitt A.3) bewertet.

Ein Vertrauensverhältnis wird durch drei Eigenschaften charakterisiert:

1. Der Gegenstand des Vertrauens (Person und Kontext), der das Vertrauensverhältnis identifiziert.
2. Der Zeitpunkt, zu dem dieses Vertrauensverhältnis gilt.
3. Der zu diesem Zeitpunkt gültige Vertrauenswert, der das Vertrauensverhältnis qualifiziert.

Der Vertrauensgegenstand, also die Identifikation einzelner Vertrauensverhältnisse, wird in unserem System durch die Klasse `TrustIdentificator` (siehe Abschnitt A.2.1) realisiert. Der Vertrauenszeitpunkt ist implizit durch die aktuelle Zeit gegeben. Der Vertrauenswert wird durch Instanzen der Klassenhierarchie `TrustValue` beschrieben (siehe Abschnitt A.2.2).

Ein Vertrauensverhältnis hat eine dynamische Natur, d.h. ein Vertrauensgegenstand kann im Laufe der Zeit unterschiedliche Vertrauenswerte zugeordnet bekommen. Folgende Situationen sind möglich:

- Zu einem Vertrauensgegenstand gibt es noch kein bekannten Vertrauenswert, doch die Information zu diesem Vertrauensverhältnis wird explizit angefordert. So muss ein initialer Vertrauenswert berechnet werden (*Vertrauensinitialisierung* in Abschnitt A.2.4).

- Bestimmte Ereignisse können möglicherweise ein oder mehrere Vertrauensverhältnisse beeinflussen. So muss das Ereignis an einen Vertrauensberechnungsalgorithmus weitergeleitet werden, der das Ereignis verarbeiten kann (*Vertrauensaktualisierung* in A.2.5).
- Ein Vertrauensverhältnis kann sich eventuell auch ganz ohne Ereignisse verändern, z.B. wenn das Vertrauen sich mit der Zeit „abnutzen“ soll (siehe 5.2.2). So muss es möglich sein zu dem Zeitpunkt einer Vertrauensabfrage diesen neu zu berechnen (*Vertrauenswertrückgabe* in Abschnitt A.2.6).

Wie jede dieser drei Situationen realisiert wird, ist jedoch nicht festgeschrieben. Sie werden durch austauschbare Vertrauensalgorithmen realisiert. Ein Vertrauensalgorithmus (siehe Abschnitt A.2.7) ist also eine Realisierung einer oder mehrerer dieser drei Berechnungssituationen. Damit die Verarbeitung an die richtigen Stellen geleitet werden kann, muss sich ein Vertrauensalgorithmus für die eine oder andere Aufgabe explizit anmelden (registrieren).

Zentrale Kontrolle der Verarbeitung der Ereignisse, Registrierung und Aufruf der geeigneten Vertrauensalgorithmen und Rückgabe der Vertrauensinformationen hat der *Vertrauensmanager* (siehe Abschnitt A.2.3).

Im Folgenden werden die hier eingeführten Punkte detaillierter beschrieben.

A.2.1. Vertrauensidentifikatoren

Ein Vertrauensverhältnis wird immer zwischen einem lokalen und einem fremden Pseudonym aufgebaut. Außerdem gilt ein Vertrauensverhältnis in einer bestimmten Kategorie zu einem bestimmten Vertrauensziel (z.B. „Weiterleitung“ oder „Empfehlung“). Diese drei Elemente (Zielpseudonym, Kategorie und Vertrauensziel) identifizieren ein Vertrauensverhältnis und werden dazu in der Klasse `TrustIdentifier` zusammengefasst.

com.hp.unitec.trust

public class **TrustIdentifier**

Methoden

String	**getCategoryId** ()	Gibt den Identifikator für die Kategorie, in der das Vertrauensverhältnis gilt zurück.
void	**setCategoryId** (String categoryId)	Setzt den Identifikator für die Kategorie, in der das Vertrauensverhältnis gilt.
String	**getPseudonymId** ()	Gibt den Identifikator für das fremden Pseudonym, der als Ziel des Vertrauensverhältnisses gilt zurück.
void	**setPseudonymId** (String pseudonymId)	

		Setzt den Identifikator für das fremden Pseudonym, der als Ziel des Vertrauensverhältnisses gilt.
String	**getPurpose** ()	Gibt das Vertrauensziel des Vertrauensverhältnisses (*recommendation* oder *forwarding*) zurück.
void	**setPurpose** (String purpose)	Setzt das Vertrauensziel des Vertrauensverhältnisses (*recommendation* oder *forwarding*).

Viele der im Folgenden beschriebenen Funktionen arbeiten mit Instanzen dieser Klasse, um eindeutig den Gegenstand des Vertrauensverhältnisses zu identifizieren.

A.2.2. Vertrauenswerte

Ein Vertrauensverhältnis wird durch einen Vertrauenswert quantifiziert. Dieser enthält jeweils eine Metrik für das Vertrauen und eine für die Vertrauensgewissheit. Vertrauenswerte werden durch Objekte aus der `TrustValue`-Hierarchie dargestellt. Die Abbildung A.3 zeigt diese Hierarchie mit einigen Beispielimplementierungen.

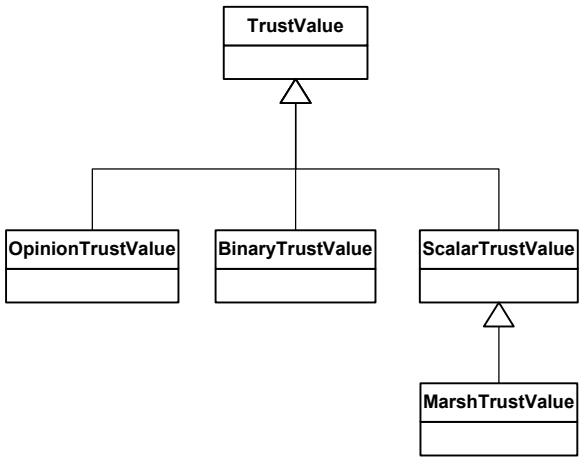


Abbildung A.3.: Vertrauenswerte in `unitec.trust`

Auf der obersten Ebene der Hierarchie (Klasse `TrustValue`) sind die Vertrauenswerte mit der Metrik des generischen Vertrauensmodells (mit der Vertrauensmetrik wie in Abschnitt 5.1.1 und der Gewissheit mit der Metrik wie in Abschnitt 5.1.2 beschrieben).

Alle weitere Vertrauensmetriken sind abgeleitete Klassen von `TrustValue`, und müssen daher die folgenden gemeinsamen Methoden implementieren.

com.hp.unitec.trust

public class **TrustValue** implements Comparable, Serializable

Methoden

void	**setTrust** (double value) throws IllegalArgumentException	Setzt die Vertrauenskomponente des Vertrauenswertes ([0, 1]).
double	**getTrust** ()	Gibt die Vertrauenskomponente des Vertrauenswertes zurück ([0, 1]).
void	**setConfidence** (double value) throws IllegalArgumentException	Setzt die Gewissheitskomponente des Vertrauenswertes ([0, 1]).
double	**getConfidence** ()	Gibt dir Gewissheitskomponente des Vertrauenswertes zurück ([0, 1]).
void	**setTrustValue** (TrustValue other)	Initialisiert diesen Vertrauenswert basierend auf ein anderen `TrustValue`.
void	**setTrustValue** (double trust, double confidence) throws IllegalArgumentException	Setzt gleichzeitig die Vertrauens- und die Gewissheitskomponente.

Um ein lokales Vertrauensmodell zu unterstützen muss eine abgeleitete Vertrauenswertklasse außerdem weitere Methoden implementieren (meistens in Form von „Getter“ und „Setter“), um den Zugriff auf die lokalen Metriken zu ermöglichen. In diesen Methoden müssen intern die Methoden der Oberklasse `TrustValue` aufgerufen werden. Dort werden also die Umrechnungen der Vertrauenswerte durchgeführt (vom generischen zum lokalen Vertrauensmodell im „Getter“ und umgekehrt im „Setter“).

Außerdem implementiert jeder `TrustValue` die `Comparable` Schnittstelle, womit unterschiedliche Vertrauenswerte geordnet werden können.

A.2.3. Vertrauensmanager

Der Vertrauensmanager ist die zentrale Steuerungseinheit der Vertrauenskomponente. Er bietet zum einem eine Schnittstelle für Klienten zur Abfrage der Vertrauensinformationen, zum anderen eine Schnittstelle zur Anmeldung von Vertrauensalgorithmen an. Intern leitet er die Informationen von außen (Ereignisse) an die entsprechenden Vertrauensalgorithmen, und die Informationen von innen (Vertrauensinformationen) auf kontrollierte Weise zurück nach außen, wie in Abbildung A.4 zu erkennen ist.

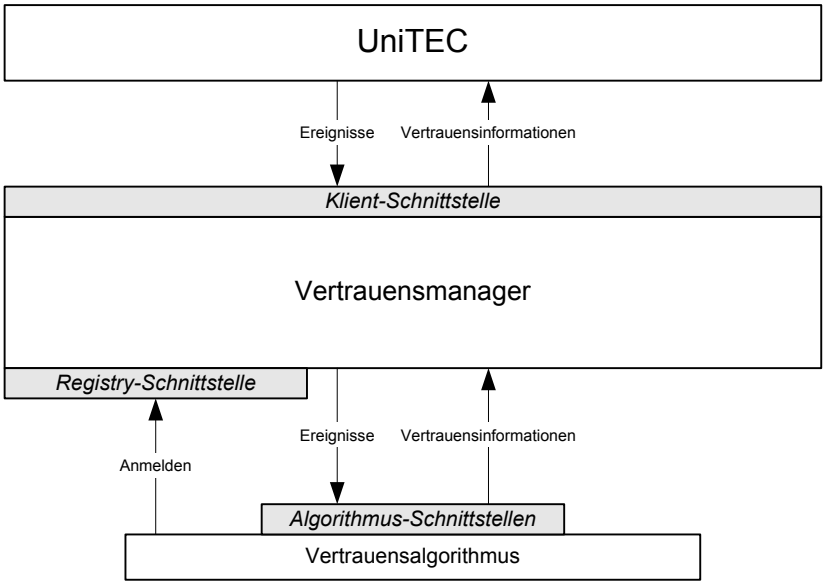


Abbildung A.4.: Vertrauensmanager

Diese Architektur sieht folgende Schnittstellen vor:

Klient-Schnittstelle Bietet jeder anderen Komponente in UniTEC die Möglichkeit Informationen zu den Vertrauensverhältnissen abzufragen oder Ereignisse zur Verarbeitung durch die Vertrauenskomponente einzureichen (`TrustClientInterface`).

Registry-Schnittstelle Ermöglicht den Vertrauensalgorithmen sich als solche anzumelden und weitere administrative Aufgaben zu erfüllen (`TrustAlgorithmManagerInterface`).

TrustManager

Die Klasse `TrustManager` ist die Fabrik-Klasse, die Implementierungen dieser Schnittstellen (Klient und Registry-Schnittstellen) anbietet.

com.hp.unitec.trust

public class **TrustManager**

Methoden

static TrustClientInterface

getTrustClientInterface ()
Gibt das Objekt zurück, das die `TrustClientInterface` Schnittstelle implementiert.

static TrustAlgorithmManagerInterface
getTrustAlgorithmManagerInterface ()
Gibt das Objekt zurück, das die `TrustAlgorithmManagerInterface` Schnittstelle implementiert.

Diese Schnittstellen werden im Folgenden genauer beschrieben.

Schnittstelle TrustClientInterface

An diese Schnittstelle wenden sich alle UniTEC-Komponenten, die Vertrauensinformationen brauchen oder die Ereignisse verursachen, die das Vertrauen verändern könnten.

com.hp.unitec.trust

public interface **TrustClientInterface**

Methoden

TrustValue **getTrustValue** (TrustIdentifier ti)
Holt den aktuell gültigen Vertrauenswert zu einem `TrustIdentifier`.

List **getMostTrusted** (String categoryId, String purpose, int maxAmount)
Gibt eine Liste fremder Pseudonyme zurück, denen in diesem Kontext am stärksten vertraut wird.

void **handleEvent** (TrustUpdateEvent tue)
Wenn ein Vertrauensereignis stattfindet, soll diese Methode aufgerufen werden, womit der Vertrauensmanager das Ereignis an die Algorithmen leiten kann, die sich für diesen Ereignistyp angemeldet haben.

void **addTrustValueChangeListener** (TrustValueChangeListener tuel, TrustIdentifier ti)
Ein Interessent registriert hier sein Wunsch über Änderungen am Vertrauenswert eines bestimmten Vertrauensidentifikators `ti` informiert zu werden. Sobald sich der Vertrauenswert ändert, wird ein Ereignis an die hier registrierten Listeners verschickt.

Als Beispiel der Anwendung dieser Schnittstelle wird hier der Vertrauenswert zu einem bestimmten `trustIdentifier` abgefragt:

Beispielcode

```
TrustClientInterface tci = TrustManager.getTrustClientInterface();
TrustValue tv = tci.getTrustValue(trustIdentifier);
...
```

TrustValueChange Ereignis Das Ereignis `TrustValueChangeEvent` kann z.B. für eine GUI-Komponente nützlich sein, die den Verlauf eines Vertrauenswertes graphisch darstellen möchte. Dazu meldet sie sich mittels `addTrustValueChangeListener` an und bekommt ab dann die Änderungen am Vertrauenswert mitgeteilt. Die GUI-Komponente kann dann z.B. den Verlauf des Vertrauenswertes graphisch nachvollziehen, ohne dass sie regelmäßig der Vertrauenswert abfragen muss. Eine Klasse, die sich mittels `addTrustValueChangeListener` registriert, muss die Schnittstelle `TrustValueChangeListener` implementieren.

com.hp.unitec.trust

public interface **TrustValueChangeListener**

Methoden

void **handle** (TrustValueChangeEvent event)
Hier kommt das Vertrauenswertänderungsereignis an. Im `TrustValueChangeEvent` sind Getter für die Eigenschaften TrustValue und TrustIdentifier vorhanden.

Schnittstelle TrustAlgorithmManagerInterface

Vertrauensalgorithmen brauchen eine Möglichkeit, sich aktiv als Verarbeiter von Vertrauensinformationen im UniTEC anzumelden. Außerdem brauchen Sie eine Möglichkeit die Funktionalität der registrierten `TrustInitialiser` und `TrustGatherer` zu nutzen. Dazu steht diese Schnittstelle zur Verfügung.

com.hp.unitec.trust

public interface **TrustAlgorithmManagerInterface**

Methoden

void **registerTrustInitialiser** (TrustInitialiser ti)
Ein Vertrauensalgorithmus kann sich hier zur Vertrauensinitialisierung anbieten. Dazu muss er selbst die Schnittstelle zur Vertrauensinitialisierung implementieren. Falls sich hier kein Algorithmus anmeldet, gilt dazu die Klasse `GenericTrustInitialiser`. Details hierzu in Abschnitt A.2.4.

void **registerTrustGatherer** (TrustGatherer tg)
Ein Algorithmus kann hier seinen eigenen Algorithmus anbieten, der die Schnittstelle zur Vertrauenswertrückgabe implementiert. Falls sich hier kein Algorithmus anmeldet, gilt dazu die Klasse `GenericTrustGatherer`. Details hierzu in Abschnitt A.2.6.

void **addTrustUpdateEventListener** (TrustUpdateEventListener tuel, Class eventClass, Map eventParameters)
Ein Algorithmus registriert sich hier, um bestimmte Ereignisse zur Verarbeitung mitgeteilt zu bekommen. `eventClass` beschreibt den spezifischen Ereignistyp, den dieser Algorithmus verarbeiten will. Nur Ereignisse dieser Klasse werden an den angemeldeten Algorithmus geleitet. Eventuelle Parameter zu ein bestimmten Ereignisstyp werden über den Parameter `eventParameters` übergeben. Details hierzu in Abschnitt A.2.5.

void **initialiseTrust** (TrustIdentifier ti)
Initialisiert den Vertrauenswert eines bestimmten `TrustIdentifier` mit Hilfe des registrierten `TrustInitialisers`.

TrustValue **getTrustValue** (TrustIdentifier ti)
Holt den aktuell gültigen Vertrauenswert zu einen `TrustIdentifier` mittels des aktuell registrierten `TrustGatherers`.

void **unregisterAlgorithm** (TrustAlgorithm ta)
Erlaubt es einem Vertrauensalgorithmus sich abzumelden. Alle Schnittstellen, die dieser Algorithmus implementiert und im Vertrauensmanager registriert hat werden hiermit abgemeldet.

Im Normalfall wird der Vertrauensalgorithmus in seiner `init()` Methode eine oder mehr der Funktionalitäten dieser Schnittstelle zur Registrierung aufrufen. Als Beispiel dient folgender Auszug einer `init()` Methode einer beispielhaften Vertrauensalgorithmusklasse.

Beispielcode

```
1 public void init() {
2   super.init();
3   TrustAlgorithmManagerInterface tami =
4     TrustManager.getTrustAlgorithmManagerInterface();
5   tami.registerTrustInitializer(this);
6   tami.addTrustUpdateEventListener(this, NewRatingEvent.class, null);
7 }
```

In Zeile 5 registriert sich der Algorithmus als Vertrauensinitialisator, in Zeile 6 meldet er sein Interesse an Ereignisse „Neue Bewertung“ an.

Die Methoden `initialiseTrust` und `getTrustValue` werden durch die einzelnen Vertrauensalgorithmen bei Bedarf verwendet (siehe hierzu auch Abschnitt A.2.7).

A.2.4. Vertrauensinitialisierung

Der Vertrauenswert muss dann initialisiert werden, wenn auf ihn aktiv zugegriffen wird und noch kein Wert gespeichert ist. Dazu muss eine Klasse die Schnittstelle `TrustInitialiser` implementieren und diese an der Schnittstelle `TrustAlgorithmManagerInterface` über die Methode `registerTrustInitialiser` registrieren (Siehe Abschnitt A.2.3):

com.hp.unitec.trust

public interface **TrustInitialiser**

Methoden

void **initialiseTrust** (TrustIdentifier trustIdentifier)
Initialisiert das Vertrauensverhältnis für den Vertrauensidentifikator `trustIdentifier`.

Falls sich kein Algorithmus für diese Funktion anmeldet, wird `GenericTrustInitialiser` als Implementierung dazu verwendet. Diese initialisiert den Vertrauenswert auf das **absolute Misstrauen** (0) und der **totalen Ungewissheit** (0).

A.2.5. Vertrauensaktualisierung

Sobald durch eine Komponente von UniTEC ein Vertrauensaktualisierungsereignis ausgelöst wird, bekommt der für dieses Ereignis registrierte Vertrauensaktualisierungsalgorithmus (siehe in A.2.3, Methode `addTrustUpdateEventListener`) das Ereignis in Form eines Objektes einer Klasse, welche Unterklasse von `TrustUpdateEvent` ist.

Die Ereignisshierarchie wird in Abbildung A.5 beispielhaft dargestellt.

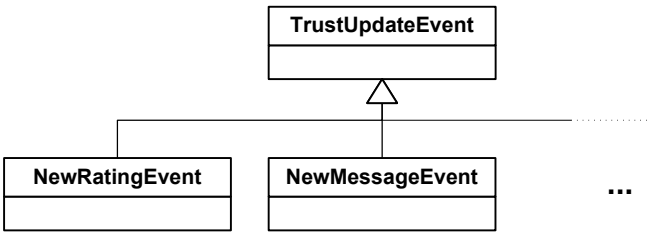


Abbildung A.5.: Vertrauensereignisse in `unitec.trust`

Eines der möglichen Ereignisse (und womöglich das wichtigste) passiert in der Bewertungskomponente beim Speichern einer neuen Bewertung. Dieses Ereignis wird durch die Klasse `NewRatingEvent` dargestellt:

com.hp.unitec.trust

public class **NewRatingEvent** extends TrustUpdateEvent

Methoden

	NewRatingEvent (Rating rhe, double weight)	Konstruktor für dieses Ereignisobjektes.
Rating	**getRating** ()	Gibt die neu gespeicherte Bewertung zurück.
double	**getWeight** ()	Gibt die Gewichtung dieser Bewertung zurück.

Die Klasse des Vertrauensalgorithmus, die ein Vertrauensaktualisierungsereignis verarbeiten will, muss die Schnittstelle `TrustUpdateEventListener` implementieren:

com.hp.unitec.trust

public interface **TrustUpdateEventListener**

Methoden

void	**handle** (TrustUpdateEvent event) throws IllegalArgumentException	Bearbeitet das Ereignis `event`. Über den Klassenname des übergebenen Ereignisobjektes (z.B. `NewRatingEvent`) kann der Algorithmus unterschiedliche Ereignisstypen voneinander unterscheiden.

Beispielsweise könnte ein Vertrauensalgorithmus diese Listener Schnittstelle folgendermaßen implementieren:

Beispielcode

```
1  public void handle(TrustUpdateEvent event) throws IllegalArgumentException {
2    if (event instanceof NewRatingEvent) {
3      NewRatingEvent newRatingEvent = (NewRatingEvent) event;
4      // ...
5      // Handle NewRatingEvent
6      // ...
7    } else {
8      throw new IllegalArgumentException(
9        "Cannot handle event " + tue.getClass().getName());
10   }
11 }
```

In Zeile 2 wird überprüft, ob es sich bei dem Ereignis um ein `NewRatingEvent` handelt.

A.2.6. Vertrauenswertrückgabe

Ein Algorithmus kann sich als Verantwortlicher zur Rückgabe des momentan gespeicherten Vertrauenswertes erklären. Somit kann er den Vertrauenswert gegebenenfalls bei einer Abfrage noch abändern. Um dies zu tun, muss die Klasse des Algorithmus die Schnittstelle `TrustGatherer` implementieren und diese bei dem TrustManager über die Methode `registerTrustGatherer` registrieren (Siehe Abschnitt A.2.3):

com.hp.unitec.trust

public interface **TrustGatherer**

Methoden

TrustValue **getTrustValue** (TrustIdentifier trustIdentifier)

Gibt den momentanen Vertrauenswert für den gewünschten Vertrauensidentifikator zurück.

Falls sich kein Algorithmus für diese Funktion anmeldet, wird `GenericTrustGatherer` als Implementierung dazu verwendet. Diese gibt lediglich den zuletzt gespeicherten Vertrauenswert zu einem Vertrauensidentifikator unverändert zurück.

A.2.7. Vertrauensalgorithmen

Ein Vertrauensalgorithmus der eine der drei Schnittstellen (`TrustInitialiser`, `TrustGatherer`, `TrustUpdateEventListener`) implementiert, sollte von der abstrakten Klasse `TrustAlgorithm`

ableiten. Dort werden einige wichtige und nützliche Funktionalitäten für jeden Vertrauensalgorithmus zur Verfügung gestellt:

- Um Vertrauenswerte abzuspeichern wird das Storage-Framework von UniTEC verwendet (siehe Details hierzu in Terdic [2003]).
- Ein Vertrauensalgorithmus muss eventuell im Laufe seiner Arbeit zu jedem Vertrauensidentifikator bestimmte algorithmusspezifische Informationen abspeichern (Zähler, Statistische Werte, usw.).

Beispielhafte Implementierungen von Vertrauensalgorithmen sind in Abbildung A.6 dargestellt. Hier werden die Vererbung- und Implementierungshierarchien deutlich.

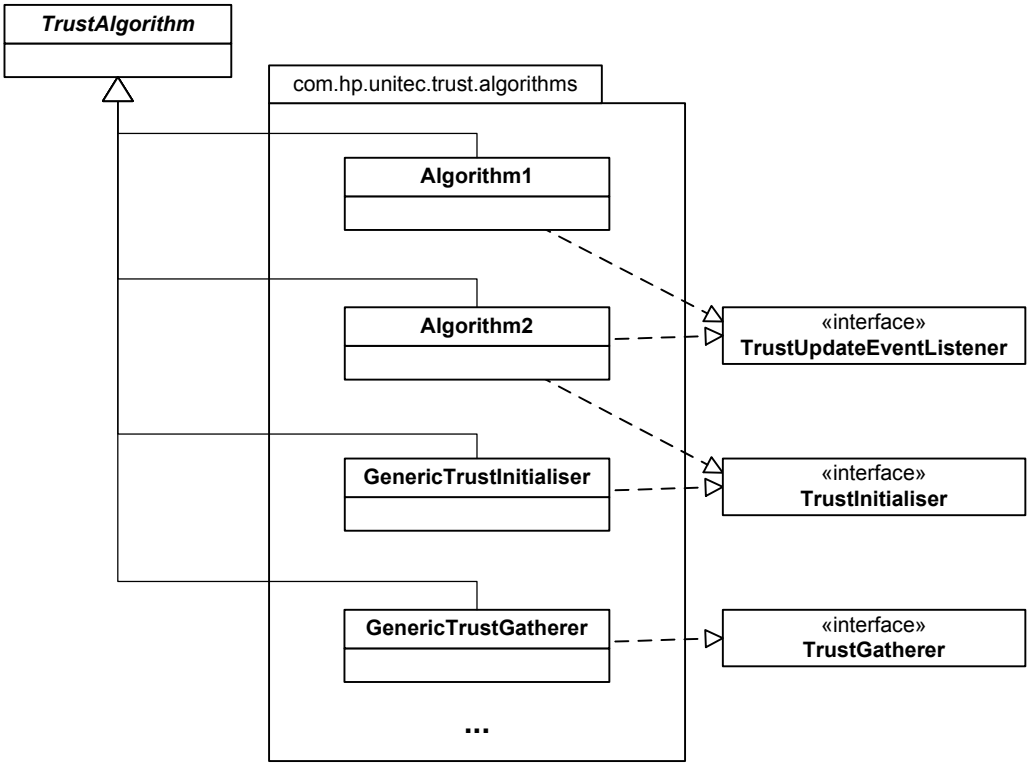


Abbildung A.6.: Implementierung von Vertrauensalgorithmen

com.hp.unitec.trust

public abstract class **TrustAlgorithm**

Methoden

Administrative Methoden

void	**init** ()	Wird aufgerufen, sobald die Klasse zum Einsatz kommen soll. Hier kann der Algorithmus entsprechend implementierte Schnittstellen bei dem Vertrauensmanager anmelden. Beispiel hierzu ist im Abschnitt A.2.3 zu finden.
void	**destroy** ()	Gegenstück zu init(), hier sollen alle Registrierungen rückgängig gemacht werden. In der Abstrakten Klasse werden bereits alle Schnittstellen am Vertrauensmanager abgemeldet. Diese Methode muss also nur überschrieben werden, wenn darüberhinaus zusätzliche Funktionalität notwendig ist.
boolean	**usesAgingFactor** ()	Der Algorithmus kann hier mitteilen, ob er Alterungsfaktoren berücksichtigen kann. Falls dies der Fall ist, muss diese Methode so überschrieben werden, dass ein wahrer Wert zurückgegeben wird.
boolean	**canWeight** ()	Der Algorithmus kann hier mitteilen, ob er unterschiedliche Gewichtungen berücksichtigen kann. Falls dies der Fall ist, muss diese Methode so überschrieben werden, dass ein wahrer Wert zurückgegeben wird.

Vertrauensspeicher

void	**initialiseTrust** (TrustIdentifier ti)	Initialisiert den Vertrauenswert eines bestimmten `TrustIdentifier`.
TrustStorageEntry	**getTrustStorageEntry** (TrustIdentifier ti)	Gibt den `TrustStorageEntry` zu den bestimmten `TrustIdentifier` zurück.
void	**storeTrustValue** (TrustIdentifier ti, TrustValue tv)	Speichert einen neuen Vertrauenswert zu einen `TrustIdentifier`.
void	**emptyTrustStorage** ()	Alle Vertrauensinformationen werden hier komplett gelöscht (Interessant beim Durchlaufen unterschiedlicher Testszenarien).

Privater Datenspeicher

private Storage
getPrivateStorage () Der Algorithmus kann sich hier einen privaten Datenspeicher holen. Dieser ist ein persistenter Datenspeicher aus dem Storage-Framework.

private Serializable
getData (TrustIdentifier ti, String itemName)
Die zuvor gespeicherten Daten können hiermit wieder ausgelesen werden.

private boolean
hasData (TrustIdentifier ti)
Wurden für diesen TrustIdentifier schon Daten gespeichert?

private void **storeData** (TrustIdentifier ti, String itemName, Serializable data)
Der Algorithmus kann sich hiermit zu einem TrustIdentifier beliebige Daten abspeichern, die später wieder unter dem entsprechenden `TrustIdentifier` und dem `itemName` gefunden werden können.

private void **emptyPrivateStorage** ()
Der private Datenspeicher wird hier komplett gelöscht (nur interessant bei Testläufen).

Bewertungspropagierung

void **handleNewRating** (NewRatingEvent nre)
Hiermit kann ein Algorithmus ohne großen Aufwand ein NewRating-Ereignis auf alle semantisch verwandten Kategorien, die diese Bewertung beeinflusst, propagieren. Um diese Funktionalität auszunutzen muss der Vertrauensalgorithmus in seiner `handle` Methode der `TrustUpdateEventListener` Schnittstelle im Falle des Ereignis `NewRatingEvent` diesen an diese Methode `handleNewRating` übergeben. Außerdem muss dann die Methode `updateTrustForNewRating` überschrieben werden, wo dann die eigentlichen Vertrauensberechnungen (jetzt pro Kategorie unterschiedlich gewichtet) durchgeführt werden.

void **updateTrustForNewRating** (TrustIdentifier ti, RatingValue rv, double weight)

Falls der Algorithmus die Propagierung einer neuen Bewertung an semantisch verwandte Kategorien mittels `handleNewRating` ausnutzt, muss sie diese Methode überschreiben und hier die Vertrauensberechnungen durchführen. Diese Methode wird jeweils einmal pro semantisch verwandter Kategorie aufgerufen.

Um algorithmusspezifische Daten zu speichern können `storeData` und `getData` genutzt werden. Falls komplexere Speichermöglichkeiten notwendig sind, bietet `getPrivateStorage` ein kompletten eigenen Speicher, der auf beliebige Art genutzt werden kann.

Die Methode `initialiseTrust` kann durch einen Vertrauenswertrückgabealgorithmus dann aufgerufen werden, wenn zu einem angefragten Vertrauensidentifikator noch kein Vertrauenswert gespeichert ist. Mit Hilfe von `storeTrustValue` und `getTrustStorageEntry` kann ein Vertrauensalgorithmus seine berechneten Vertrauenswerte abspeichern und später wieder abfragen.

Die Klasse `TrustStorageEntry` ist dabei eine Subklasse von `StorageEntry` und enthält zusätzlich zu den Möglichkeiten von `StorageEntry` folgende Methoden:

com.hp.unitec.trust

public class **TrustStorageEntry** extends StorageEntry

Methoden

TrustIdentifier	**getTrustIdentifier** ()	Gibt den Vertrauensidentifikator zurück.
void	**setTrustIdentifier** (TrustIdentifier trustIdentifier)	Setzt den Vertrauensidentifikator.
TrustValue	**getTrustValue** ()	Gibt den aktuellen Vertrauenswert zurück.
void	**setTrustValue** (TrustValue trustValue)	Setzt den aktuellen Vertrauenswert.

A.3. Bewertungskomponente, Paket `unitec.rating`

Bewertungen werden durch den Benutzer oder automatisch durch das UniTEC-System generiert und gespeichert. Diese Bewertungen beeinflussen die Vertrauenswerte, die zu einzelnen fremden Pseudonymen gespeichert sind. Wie die Bewertungen verarbeitet werden ist Sache der Vertrauensalgorithmen. Das Packet `com.hp.unitec.trust.rating` bietet lediglich die Verwaltungsmethoden für die Bewertungen an. Es bedient die Vertrauenskomponenten mit dem

Ereignis NewRatingEvent bei jeder neu gespeicherten Bewertung (siehe dazu mehr in A.2.5).

A.3.1. Bewertungswerte

Bewertungswerte werden durch Objekte aus der RatingValue-Hierarchie dargestellt. Die Abbildung A.7 zeigt diese Hierarchie.

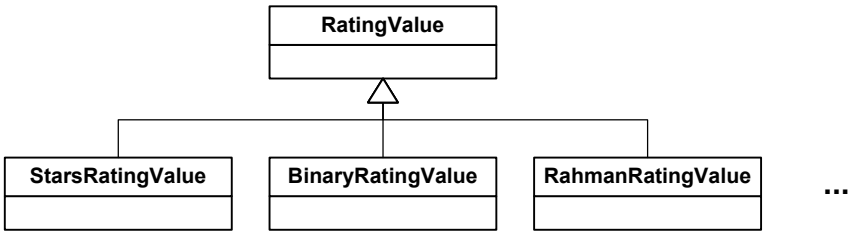


Abbildung A.7.: Bewertungswerte in unitec.trust.rating

Instanzen der Klasse RatingValue repräsentieren Bewertungswerte des generischen Vertrauensmodells (mit der Bewertungsmetrik wie in 5.1.3). Alle weiteren Bewertungsmetriken sind abgeleitete Klassen von RatingValue und implementieren daher dieselben gemeinsamen Methoden (ähnlich wie bei der Vertrauenswerte Hierarchie).

com.hp.unitec.trust.rating

public class **RatingValue** implements Comparable, Serializable

Methoden

double	**getValue** ()	Gibt den Bewertungswert zurück.
void	**setValue** (double value) throws IllegalArgumentException	Setzt den Bewertungswert ([0, 1]).

Jeder RatingValue implementiert die Comparable Schnittstelle, womit man unterschiedliche Bewertungswerte ordnen kann. Alle weiteren Bewertungswertklassen können weitere Methoden implementieren, um den Zugriff auf die eigenen Metriken zu ermöglichen. Die oben beschriebenen Methoden müssen durch abgeleitete Klassen die Funktion beibehalten, auch wenn über anderen Methoden auf die eigene Metrik zugegriffen wird.

A.3.2. Bewertungen

Einzelne Bewertungen werden als Instanzen der Klasse Rating gespeichert.

com.hp.unitec.trust.rating

public class **Rating** extends StorageEntry implements Comparable

Methoden

long	**getCreationTime** ()	Gibt den Zeitpunkt zu dem die Bewertung erzeugt wurde zurück.
void	**setCreationTime** (long creationTime)	Setzt den Zeitpunkt zu dem die Bewertung erzeugt wurde.
RatingValue	**getRatingValue** ()	Gibt den Bewertungswert dieser Bewertung zurück.
void	**setRatingValue** (RatingValue ratingValue)	Setzt den Bewertungswert dieser Bewertung.
String	**getCategoryId** ()	Gibt die Kategorie, in der diese Bewertung abgegeben wurde zurück.
void	**setCategoryId** (String categoryId)	Setzt die Kategorie, in der diese Bewertung abgegeben wurde.
String	**getRecommenderId** ()	Gibt den Empfehler, der in dieser Bewertung bewertet wird zurück.
void	**setRecommenderId** (String recommenderId)	Setzt den Empfehler, der in dieser Bewertung bewertet wird.
double	**getRecommenderCertainty** ()	Gibt die Gewissheit des Empfehlers in seine Empfehlung zurück.
void	**setRecommenderCertainty** (double recommenderCertainty)	Setzt die Gewissheit des Empfehlers in seine Empfehlung.
double	**getUtilityFactor** ()	Gibt die Wichtigkeit der Empfehlung, die in dieser Bewertung bewertet wird zurück.
void	**setUtilityFactor** (double utilityFactor)	Setzt die Wichtigkeit der Empfehlung, die in dieser Bewertung bewertet wird.
TrustChain	**getTrustChain** ()	Gibt die Vertrauenskette vom Anfragenden bis zum Empfehler zurück.
void	**setTrustChain** (TrustChain trustChain)	

		Setzt die Vertrauenskette vom Anfragenden bis zum Empfehler.
TrustIdentifier		
	getTrustIdentifier ()	Gibt den Vertrauensidentifikator zurück, der für diese Bewertung relevant ist.

A.3.3. Bewertungsmanager

Ähnlich wie bei den Kategorien und bei Vertrauen wird auch hier der Zugriff auf die Bewertung über eine einheitliche Schnittstelle (`RatingsManager`) gehandhabt.

com.hp.unitec.trust.rating

public class **RatingsManager**

Methoden

static RatingsManager		
	getEntity ()	Gibt das Singleton-Objekt des `RatingsManagers` zurück.
void	**storeRating** (Rating rating)	Speichert eine neue Bewertung. Die Bewertung wird persistent gespeichert. Anschließend wird das Ereignis `NewRatingEvent` (siehe A.2.5) ausgelöst.
Vector	**getRatings** (String pseudonymId, String categoryId)	Gibt alle Bewertungen, die über das gewünschte Pseudonym in der gewünschten Kategorie abgegeben worden sind, in chronologischer Reihenfolge zurück.

Es folgt nun ein Beispiel, wie eine neue Bewertung erzeugt und gespeichert wird.

Beispielcode

```
Rating rating = new Rating();
rating.setRatingValue(ratingValue);
rating.setRecommenderId(pseudonymId);
rating.setCategoryId(categoryId);
...
RatingsManager rm = RatingsManager.getEntity();
rm.storeRating(rating);
...
```

A.4. Test Framework, Paket `unitec.tests.trust`

Das Testframework definiert einige Komponenten, die zur Analyse von Vertrauensmodellen nützlich sein können. Es wurde z.B. für die Durchführung der Testszenarien aus Abschnitt 8.1 verwendet.

A.4.1. Testszenarien

Ein Testszenario repräsentiert eine Bewertungssequenz. Diese Sequenz wird nach und nach über die Bewertungskomponente gespeichert, und löst somit Vertrauensberechnungen durch Vertrauensaktualisierungsereignisse aus.

Alle Testszenarien sind im Packet `unitec.tests.trust.scenarios` zu finden, wobei jede Klasse die Schnittstelle `TrustTestScenario` implementieren muss. Diese Schnittstelle ist folgendermassen festgelegt:

com.hp.unitec.tests.trust

public interface **TrustTestScenario**

Methoden

void	**reset** ()	Das Testszenario wird hiermit in den initialen Zustand zurückversetzt.
boolean	**hasNextRatingValue** ()	Es wird angefragt, ob weitere Bewertungswerte zur Verfügung stehen.
RatingValue	**getNextRatingValue** ()	Falls hasNextRatingValue() true zurückgibt, kann hiermit der nächste Bewertungswert abgeholt werden.
void	**setNextTrustValue** (TrustValue tv)	Nach jeder abgegebene Bewertung wird hier zurück an das Testszenario das sich ergebene Vertrauensverhältnis vermittelt. Somit kann das Testszenario Bewertung aufgrund dieser Information abgeben (und somit eine dynamische Bewertungssequenz erzeugen).

A.4.2. Ausgabe

Während ein Testszenario durchlaufen wird, kann dessen Ergebnisse in Dateien (z.B. als CSV oder GnuPlot Dateien) geschrieben, oder auch interaktiv angezeigt werden.

Bei einer Ausführung des Testframeworks (siehe später), werden ein oder mehrere Algorithmen und ein oder mehrere Testszenarien zur Ausführung ausgewählt. Jedes Testszenario wird mehrmals durchlaufen, einmal pro ausgewählten Algorithmen (mit den ausgewählten Parametern). Jeder dieser Lauf heisst „*TestRun*“.

Eine Klasse, die die Ausgabe solcher Testläufe beschreiben möchte, soll die Schnittstelle `TestScenarioOutputListener` implementieren, die folgendermassen definiert ist:

com.hp.unitec.tests.trust

public interface **TestScenarioOutputListener**

Methoden

void	**startTestScenario** (String name)	Wird aufgerufen, wenn die Durchführung eines bestimmten Testszenarios beginnt.
void	**startRun** (String id, String name, Map parameters)	Wird aufgerufen, wenn ein Testlauf begonnen wird.
void	**runEntry** (String id, int counter, double ratingValue, double trustValue, double confidenceValue)	Nach jeder abgegebene Bewertung des Testszenarios wird diese Methode aufgerufen. Hier wird mitgeteilt, welche Bewertung abgegeben wurde und welche die sich daraus ergebene Vertrauens- und Gewissheitswerte (in den Metriken des generischem Vertrauensmodells) sind.
void	**endRun** (String id, String name, Map parameters)	Wird abgerufen, wenn der komplette Testlauf abgeschlossen ist.
void	**endTestScenario** (String name)	Wird aufgerufen, wenn die ein Testszenario komplett durchlaufen wurde.
void	**endAll** ()	Wird abgerufen, wenn die komplette Ausführung beendet ist.

Implementiert sind bereits drei solche Klassen, und zwar:

SaveGnuplotHandler Speichert die Testläufe in einzelne Dateien, die mit Hilfe von Gnuplot (vgl. Gnuplot [2004]) visualisiert werden können.

SaveCsvfilesHandler Speichert die Testläufe in CSV-Dateien (Komma-getrennte Dateien), die in gängigen Tabellenkalkulationsprogrammen geöffnet und analysiert werden können.

DisplayGraphHandler Ein bestimmter Testlauf kann hier graphisch und in Echtzeit visualisiert werden. Die Anzeige verwendet die Java-Bibliothek aus [Westermann, 2004].

A.4.3. Durchführung der Testläufe

Die beiden Schnittstellen werden bereits in zwei Anwendungen genutzt. Diese können verwendet werden, um Testszenarien durchlaufen zu lassen:

ProcessTestScenarios Hier werden alle möglichen Kombination von eine Auswahl an Algorithmen, Parameter (Alterung- und Gewichtung) und Ausgabeformate nacheinander in einem „Batch-Modus“ durchlaufen. Das Ergebnis ist abhängig von den eingesetzten `TestScenarioOutputListener`.

ProcessTestScenarioAsGraph Diese Anwendung beschränkt sich auf die Visualisierung eines bestimmten Algorithmus unter einem Testszenario. Die Visualisierung basiert auf

B. Testergebnisse

Im Folgenden werden die Ergebnisse der Testläufe der einzelnen Testszenarien dargestellt, die nicht explizit in den Analysen betrachtet wurden. Zu jedem Testszenario wird jeweils der Vertrauensverlauf unter jeden der betrachteten Vertrauensaktualisierungsalgorithmen in zwei getrennten Abbildungen graphisch dargestellt.

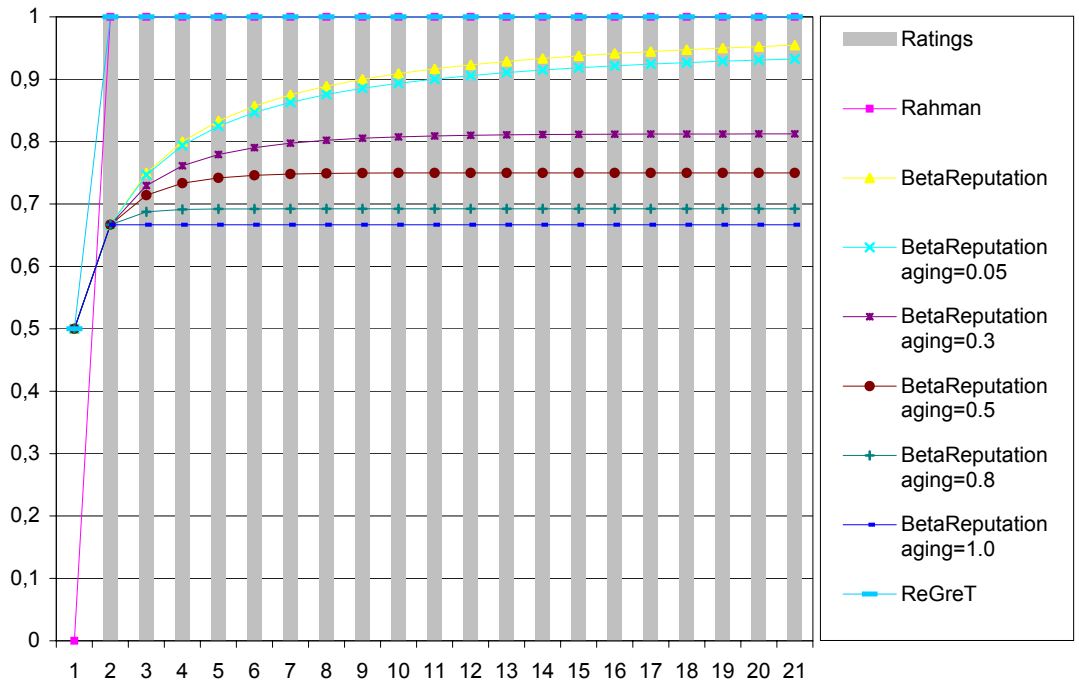


Abbildung B.1.: Vertrauensverlauf im Testszenario *OnlyMaximalRatings*, Teil 1

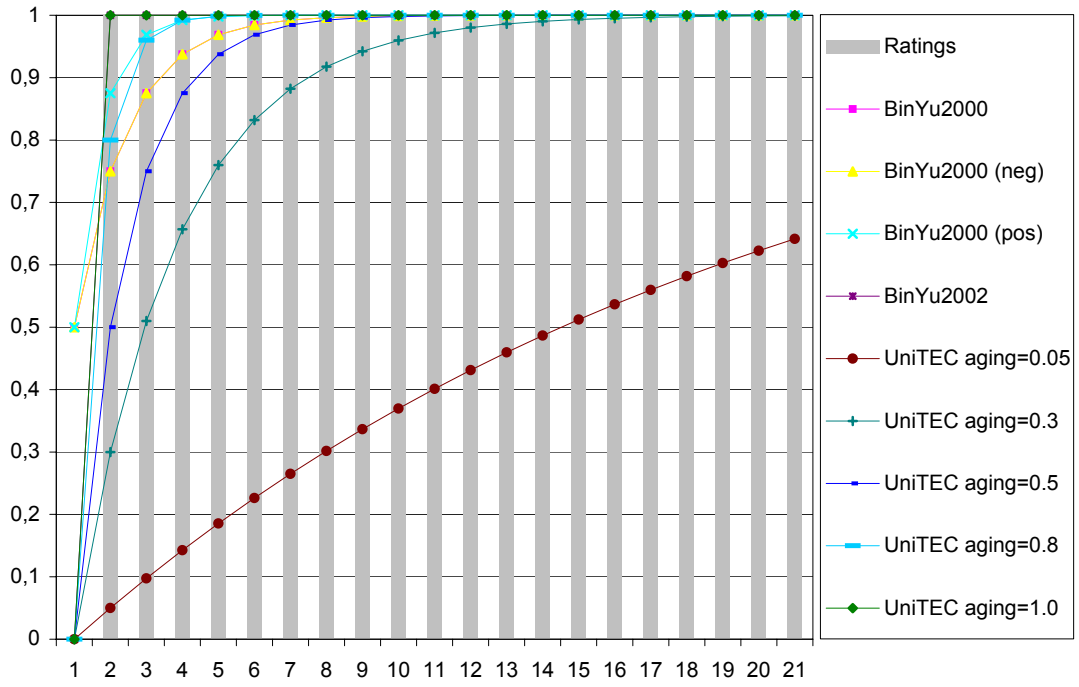


Abbildung B.2.: Vertrauensverlauf im Testszenario *OnlyMaximalRatings*, Teil 2

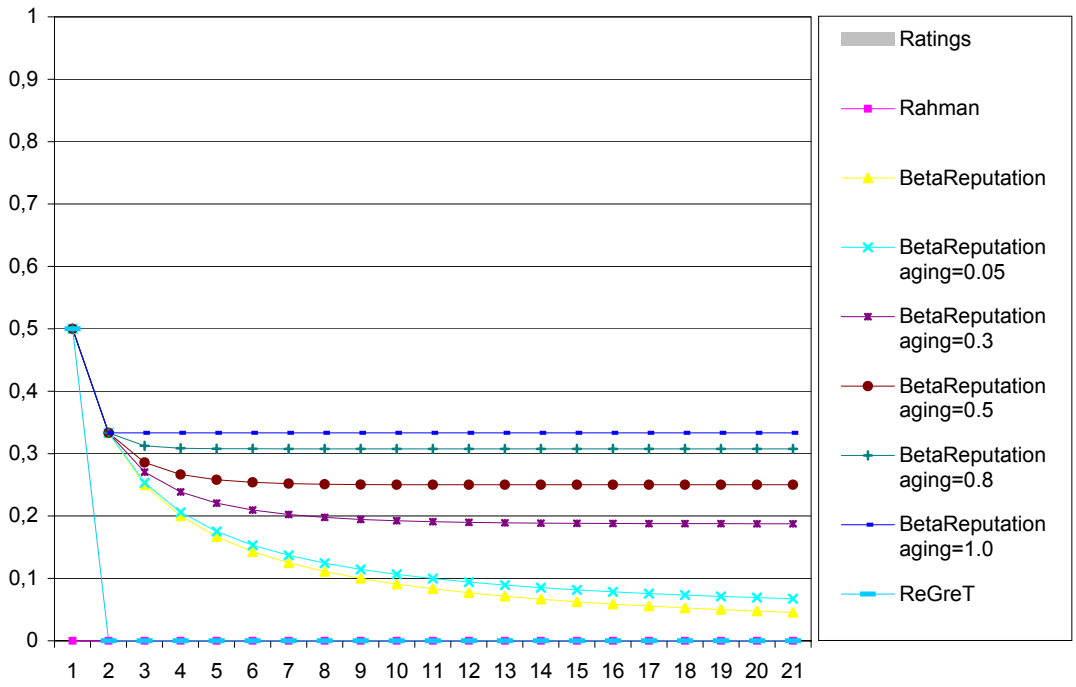


Abbildung B.3.: Vertrauensverlauf im Testszenario *OnlyMinimalRatings*, Teil 1

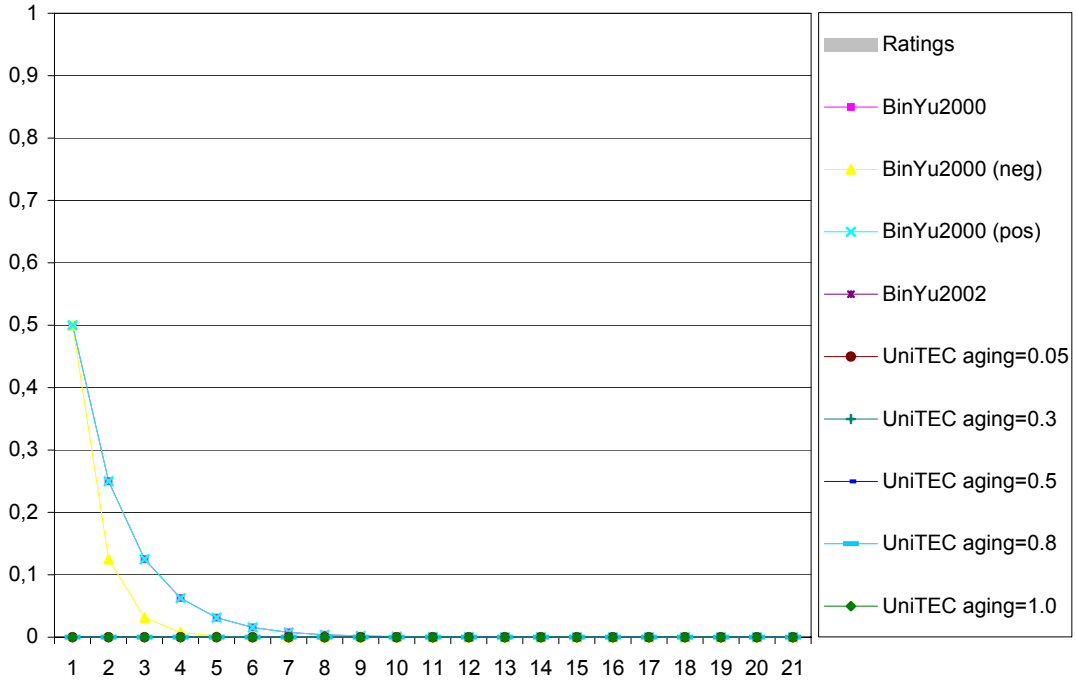


Abbildung B.4.: Vertrauensverlauf im Testszenario *OnlyMinimalRatings*, Teil 2

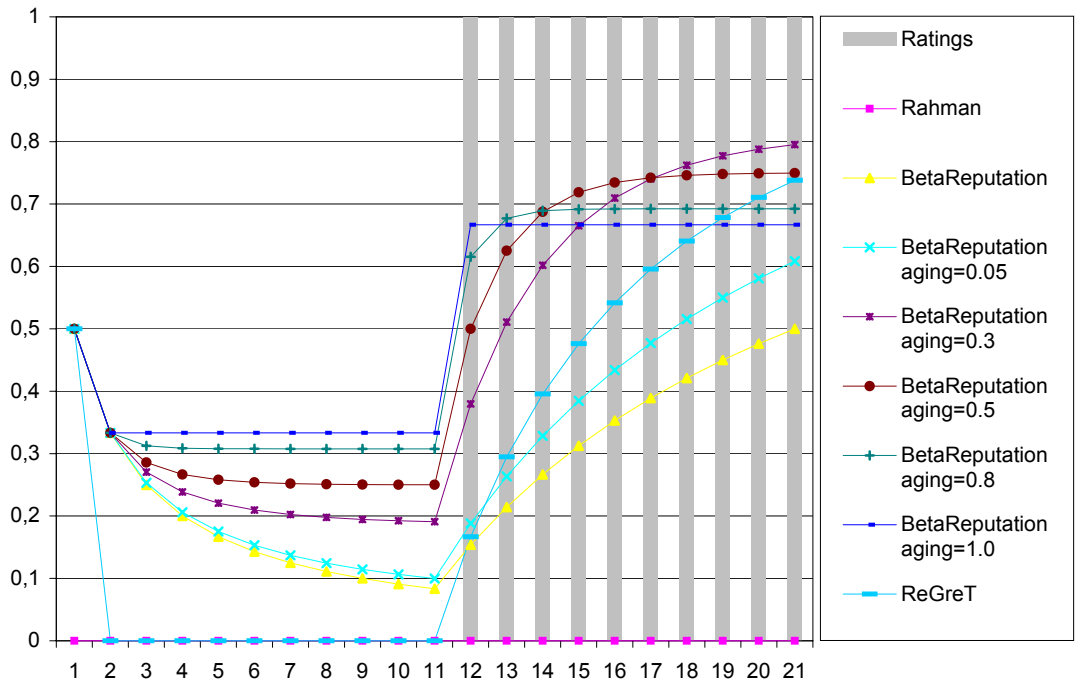


Abbildung B.5.: Vertrauensverlauf im Testszenario *MinimalThenMaximalRatings*, Teil 1

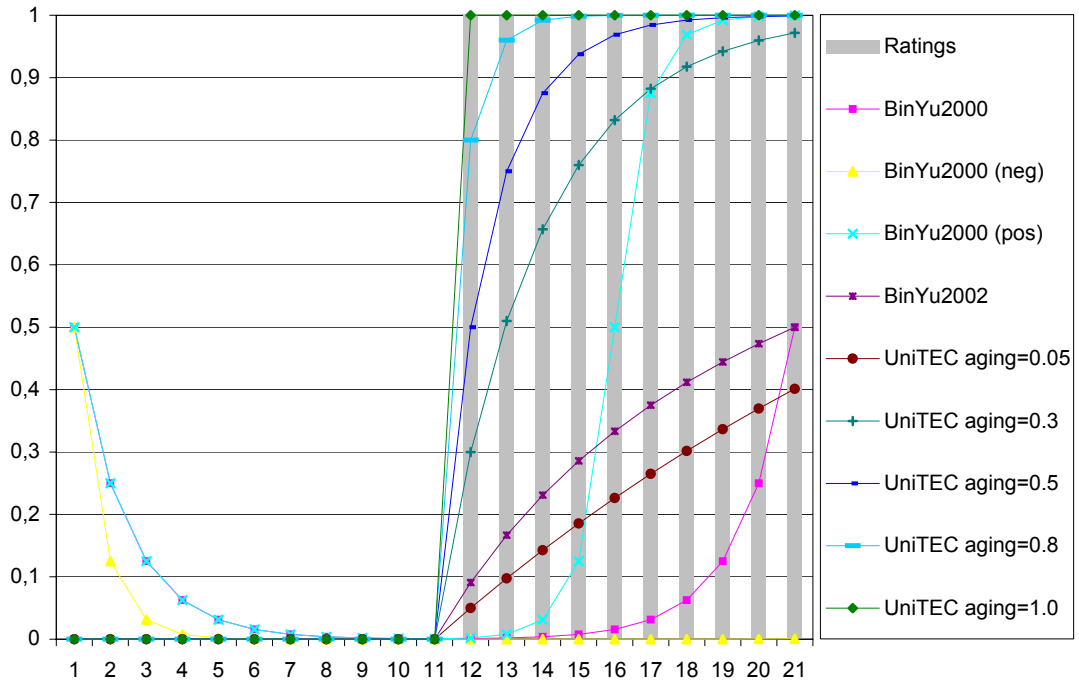


Abbildung B.6.: Vertrauensverlauf im Testszenario *MinimalThenMaximalRatings*, Teil 2

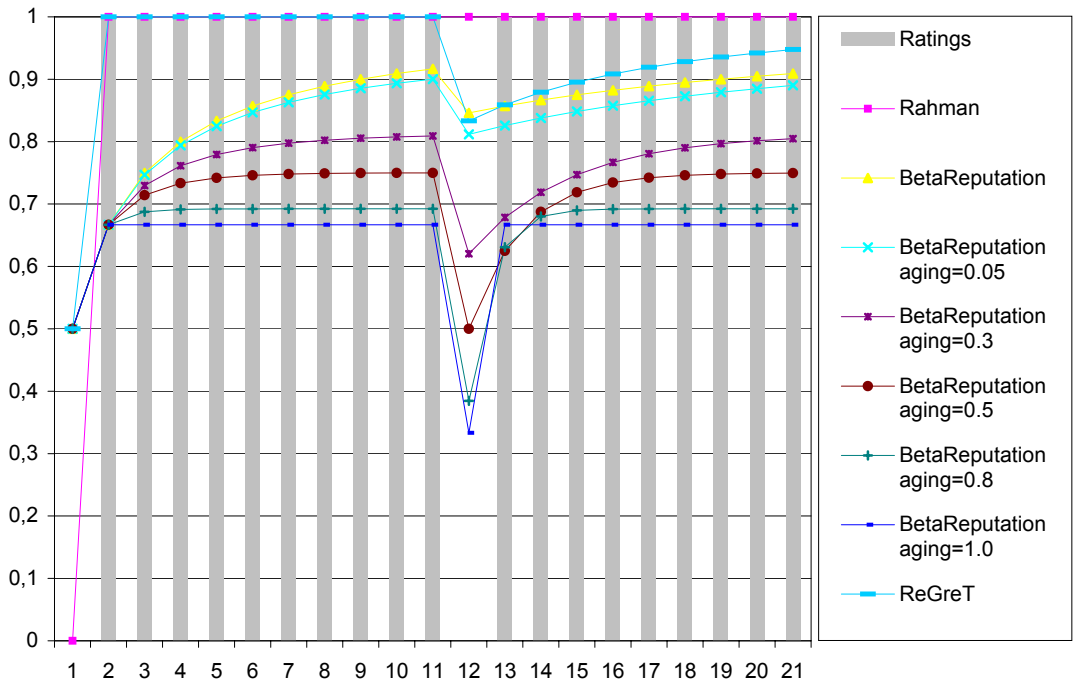


Abbildung B.7.: Vertrauensverlauf im Testszenario *MaximalInterruptedByMinimal*, Teil 1

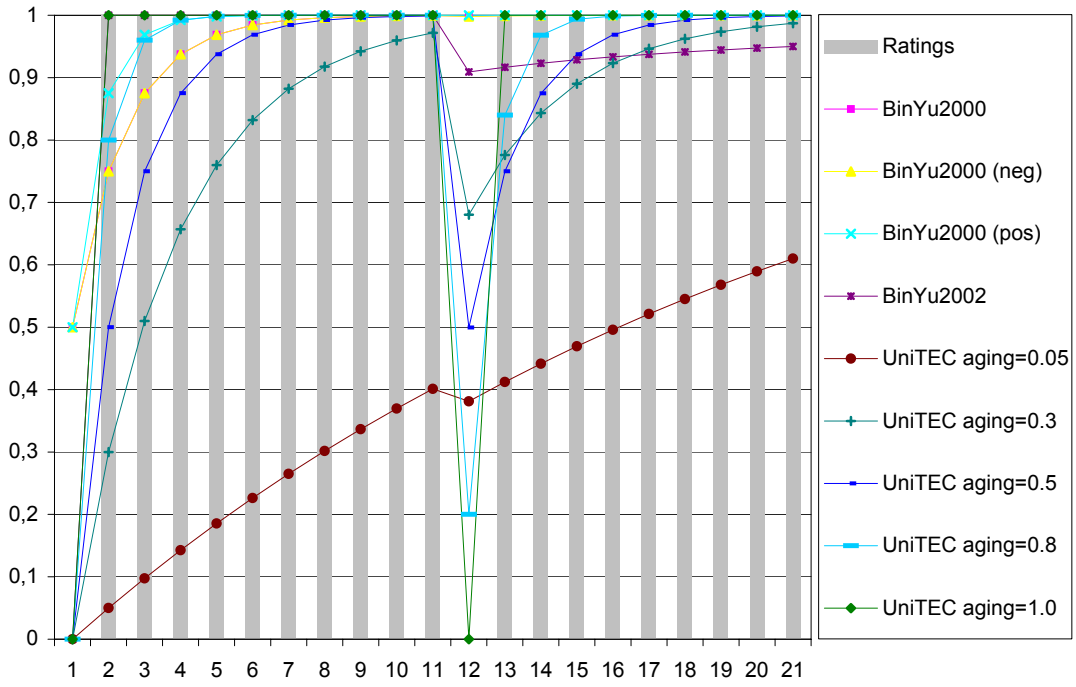


Abbildung B.8.: Vertrauensverlauf im Testszenario *MaximalInterruptedByMinimal*, Teil 2

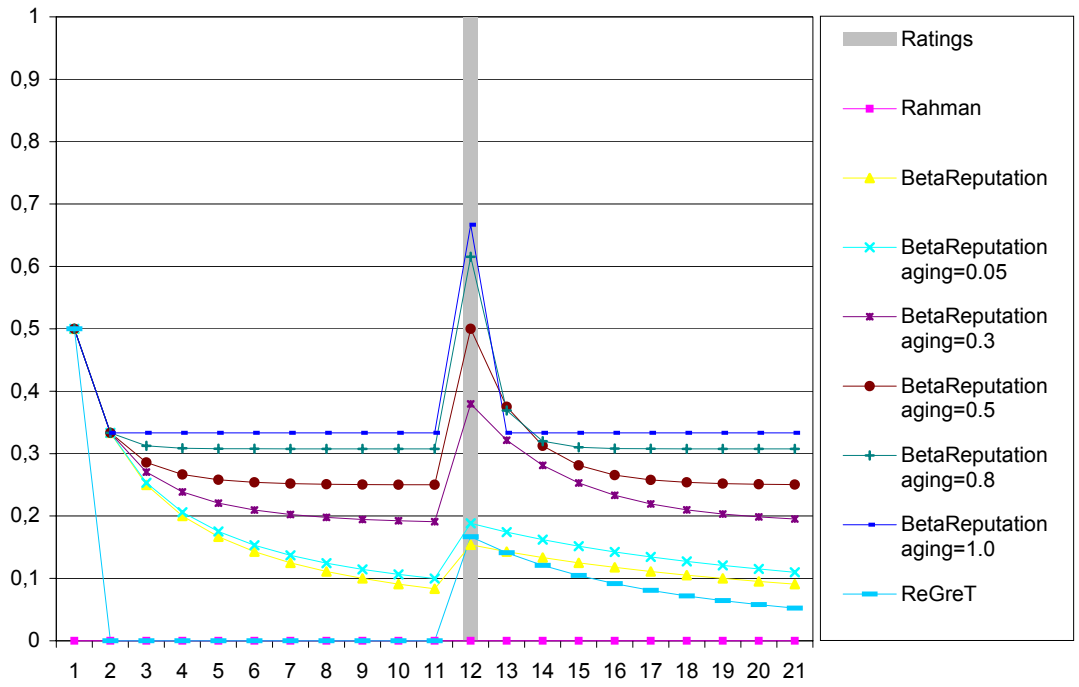


Abbildung B.9.: Vertrauensverlauf im Testszenario *MinimalInterruptedByMaximal*, Teil 1

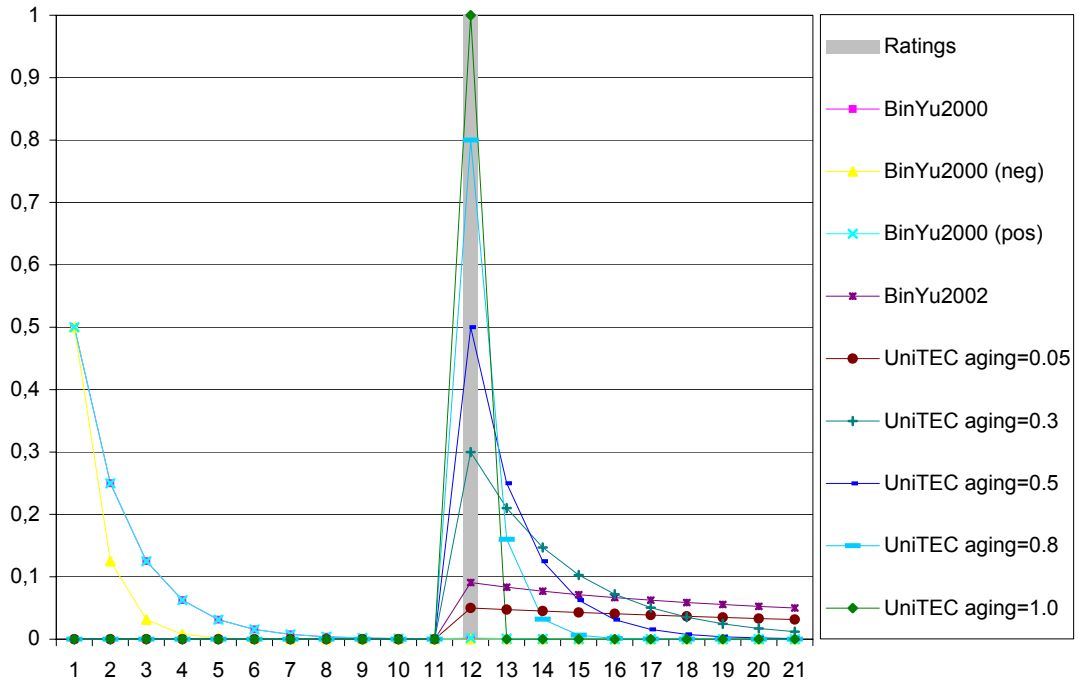


Abbildung B.10.: Vertrauensverlauf im Testszenario *MinimalInterruptedByMaximal*, Teil 2

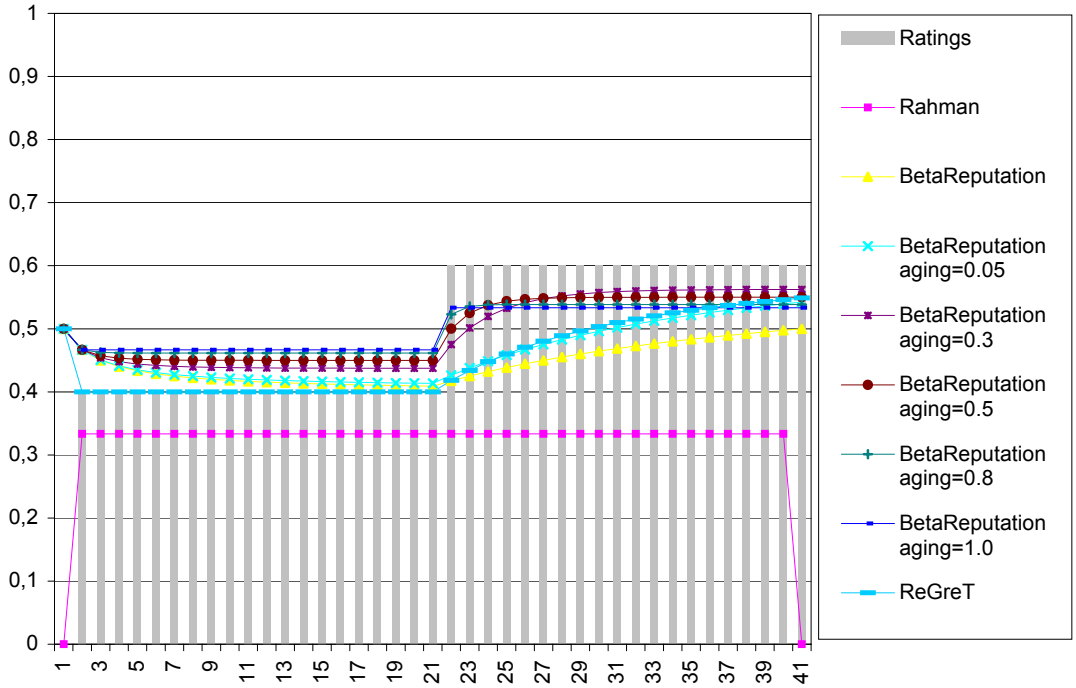


Abbildung B.11.: Vertrauensverlauf im Testszenario *SlightlyNegativeThenPositiveRatings*, Teil 1

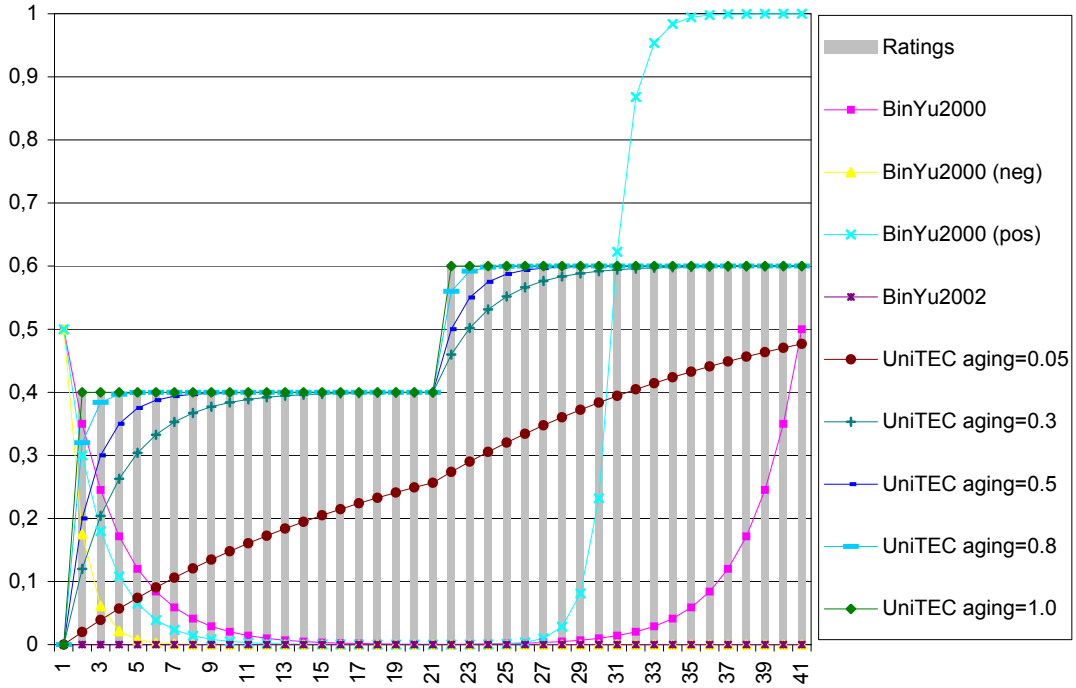


Abbildung B.12.: Vertrauensverlauf im Testszenario *SlightlyNegativeThenPositiveRatings*, Teil 2

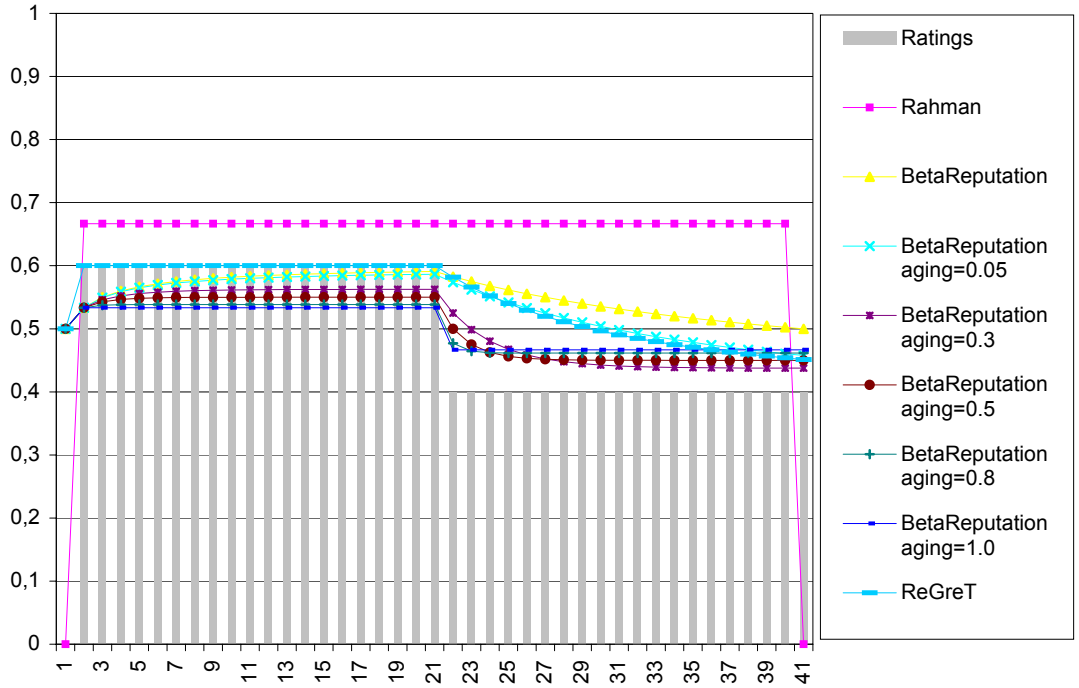


Abbildung B.13.: Vertrauensverlauf im Testszenario *SlightlyPositiveThenNegativeRatings*, Teil 1

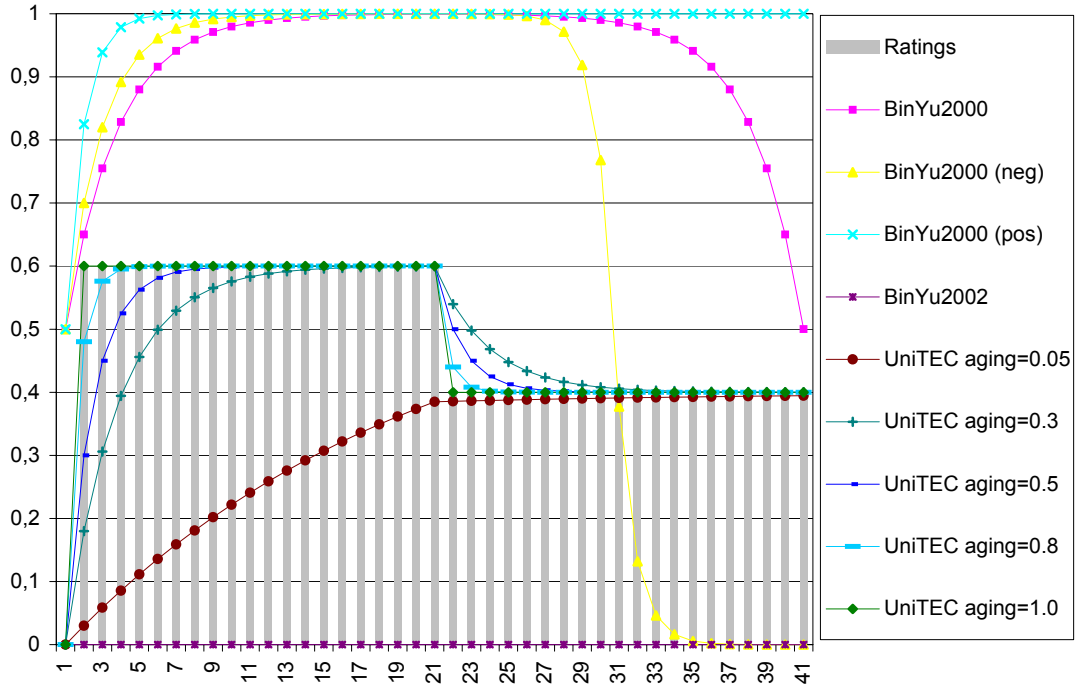


Abbildung B.14.: Vertrauensverlauf im Testszenario *SlightlyPositiveThenNegativeRatings*, Teil 2

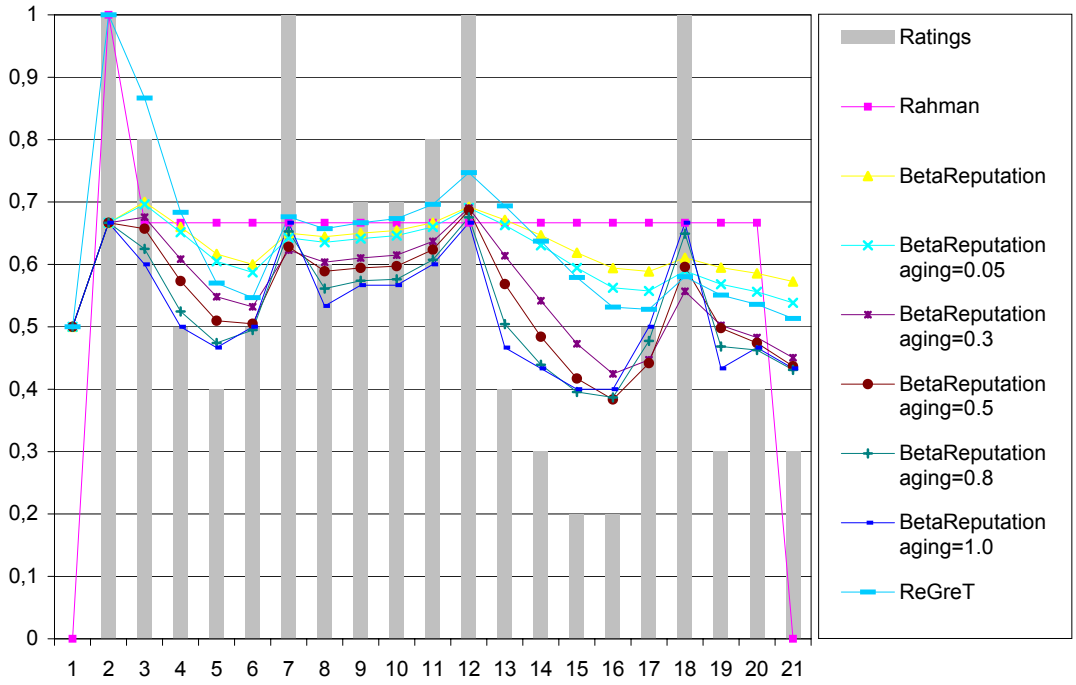


Abbildung B.15.: Vertrauensverlauf im Testszenario *SpecificRatingsNormal*, Teil 1

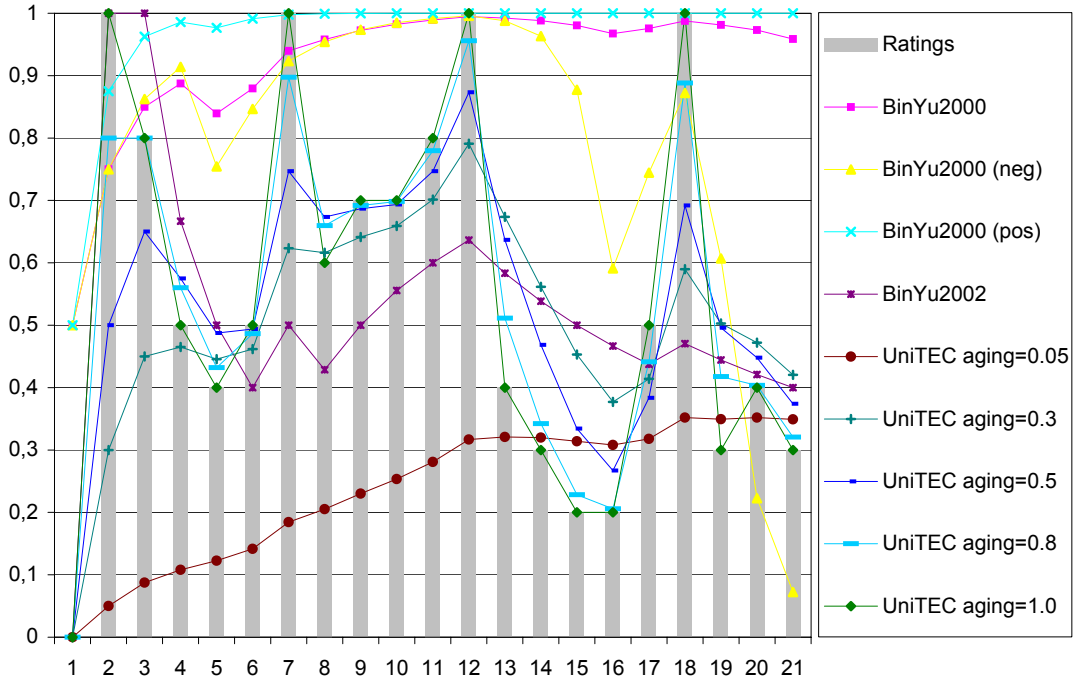


Abbildung B.16.: Vertrauensverlauf im Testszenario *SpecificRatingsNormal*, Teil 2

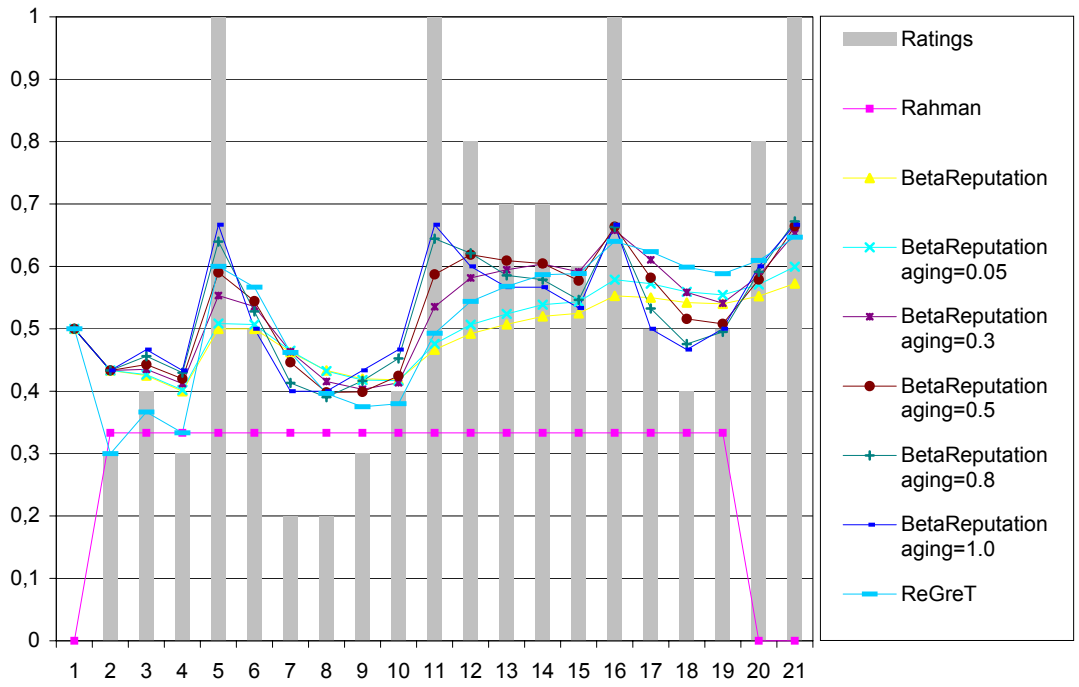


Abbildung B.17.: Vertrauensverlauf im Testszenario *SpecificRatingsReversed*, Teil 1

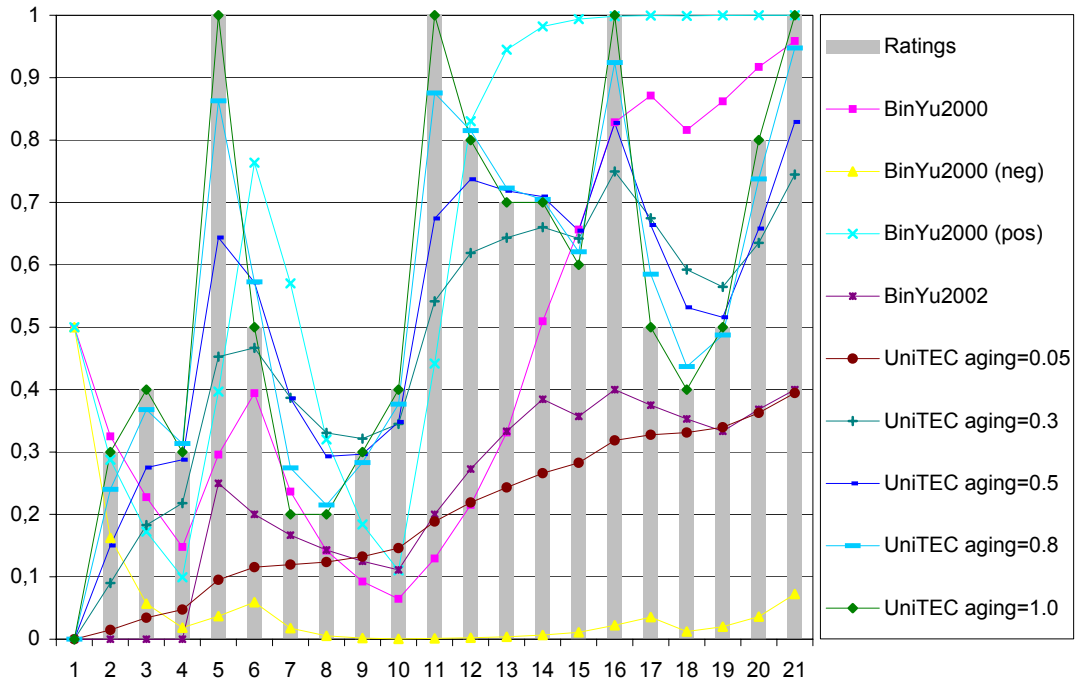


Abbildung B.18.: Vertrauensverlauf im Testszenario *SpecificRatingsReversed*, Teil 2

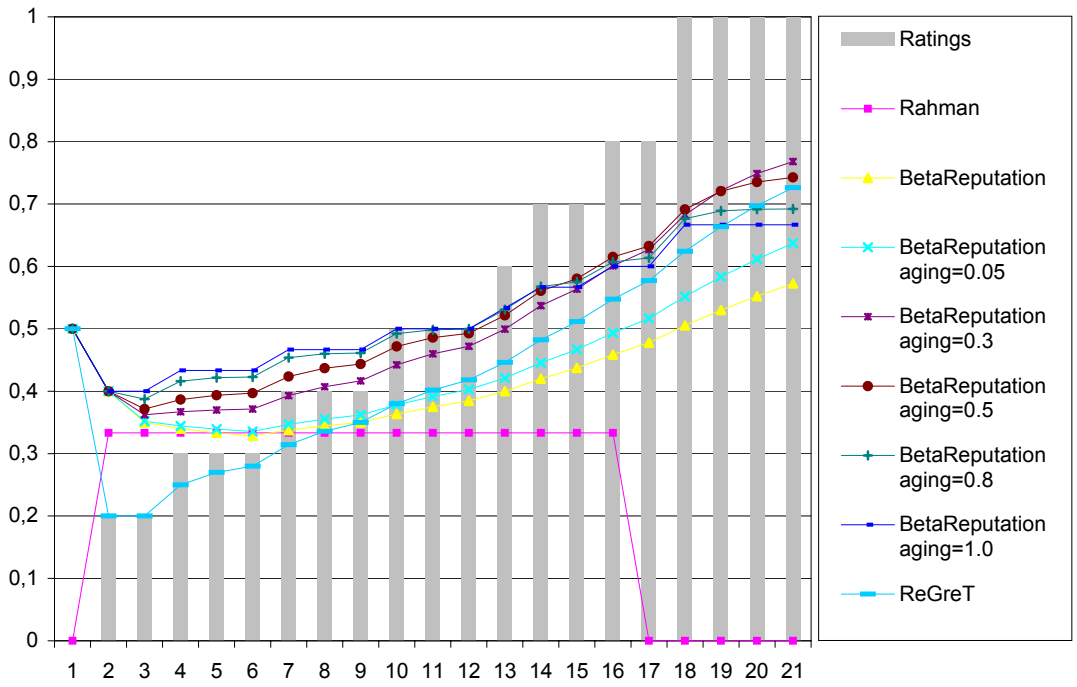


Abbildung B.19.: Vertrauensverlauf im Testszenario *SpecificRatingsOrderedAsc*, Teil 1

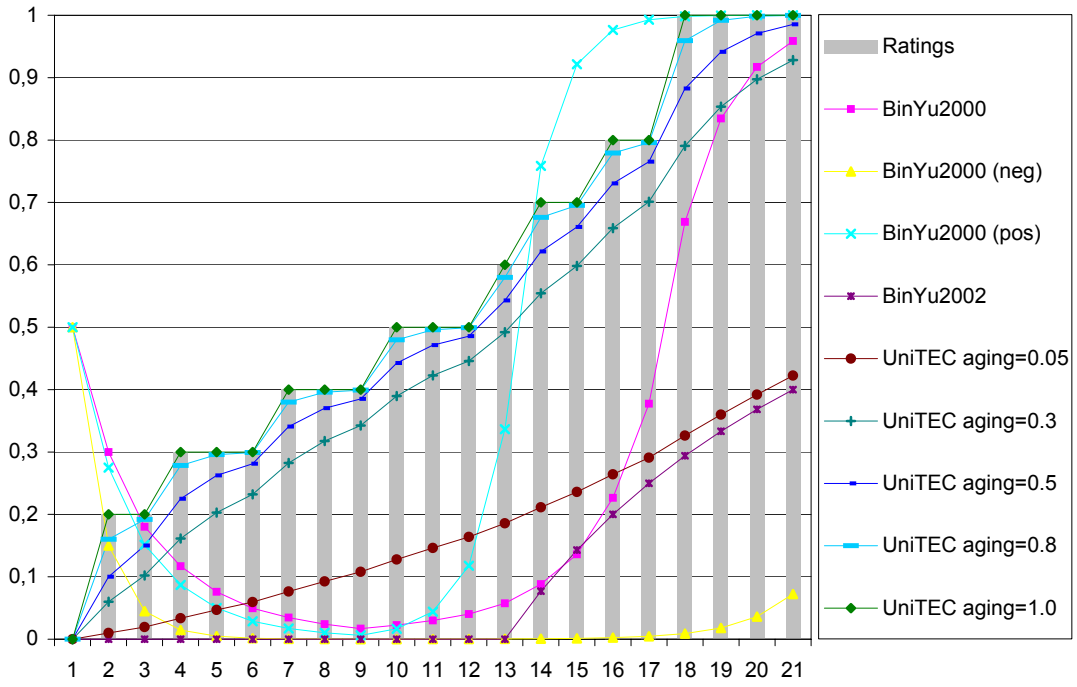


Abbildung B.20.: Vertrauensverlauf im Testszenario *SpecificRatingsOrderedAsc*, Teil 2

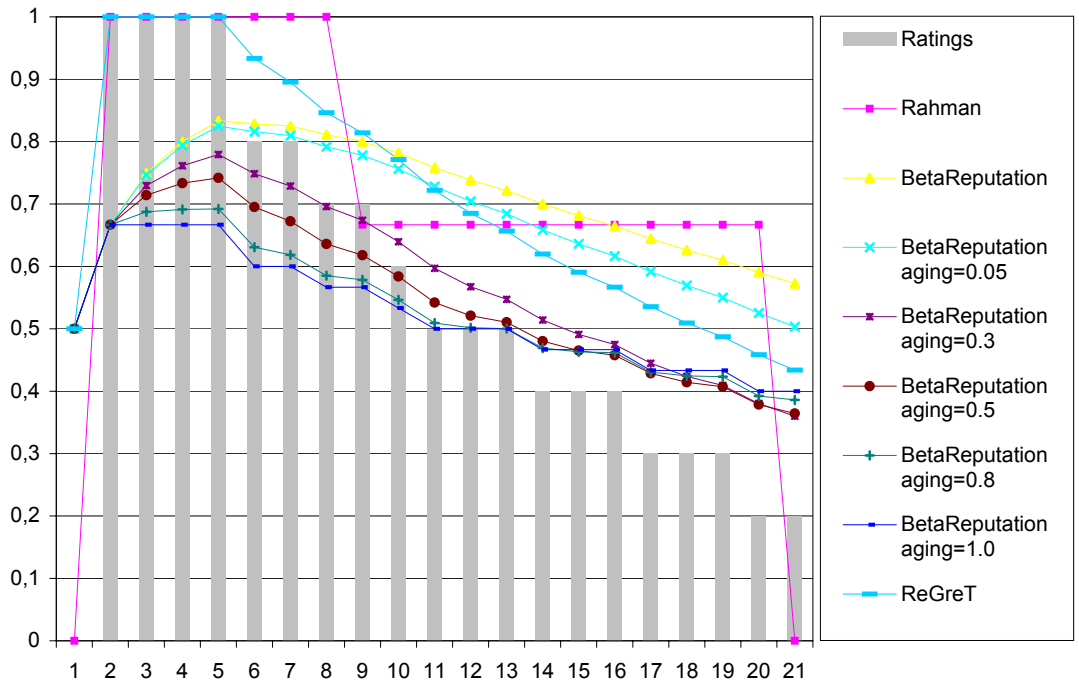


Abbildung B.21.: Vertrauensverlauf im Testszenario *SpecificRatingsOrderedDesc*, Teil 1

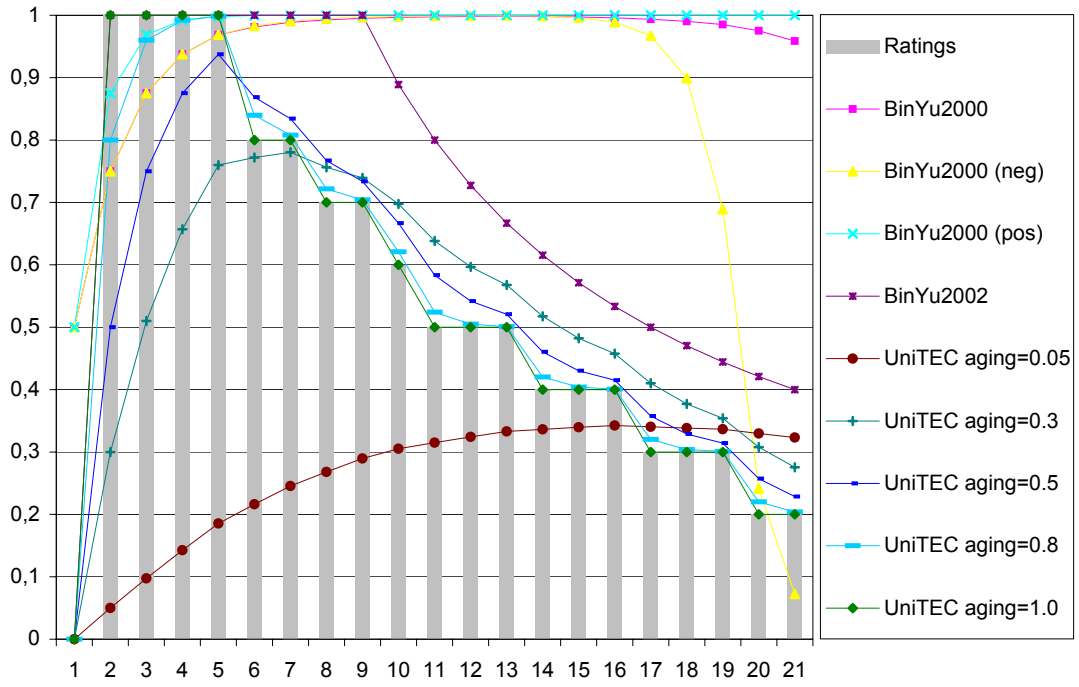


Abbildung B.22.: Vertrauensverlauf im Testszenario *SpecificRatingsOrderedDesc*, Teil 2

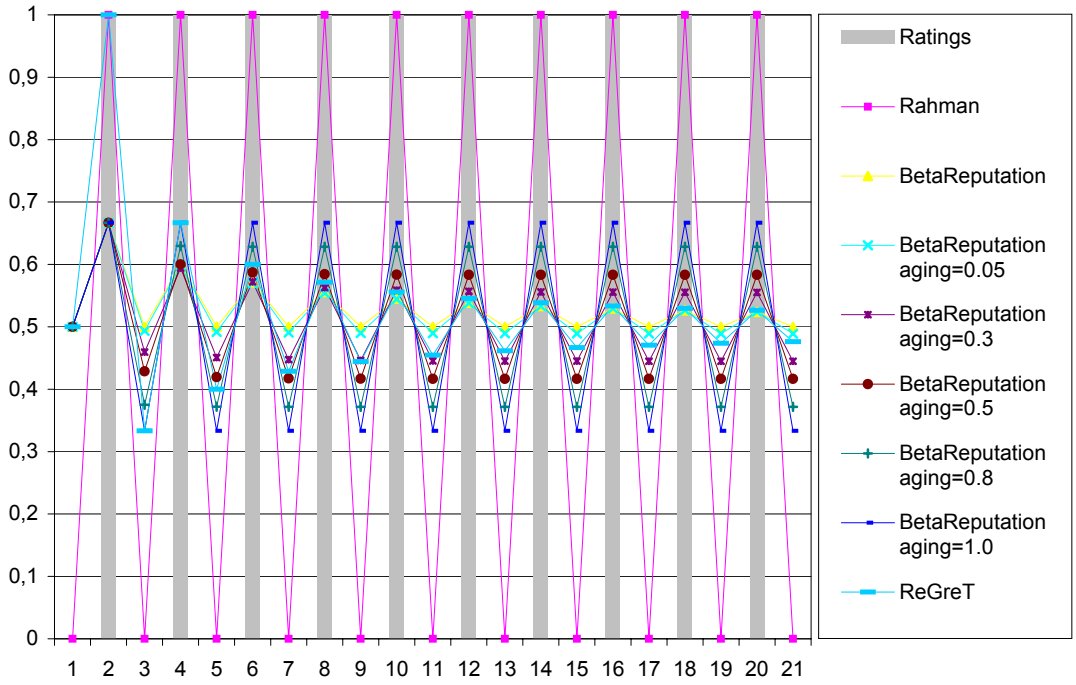


Abbildung B.23.: Vertrauensverlauf im Testszenario *OscilatingRatingsFirstMaximal*, Teil 1

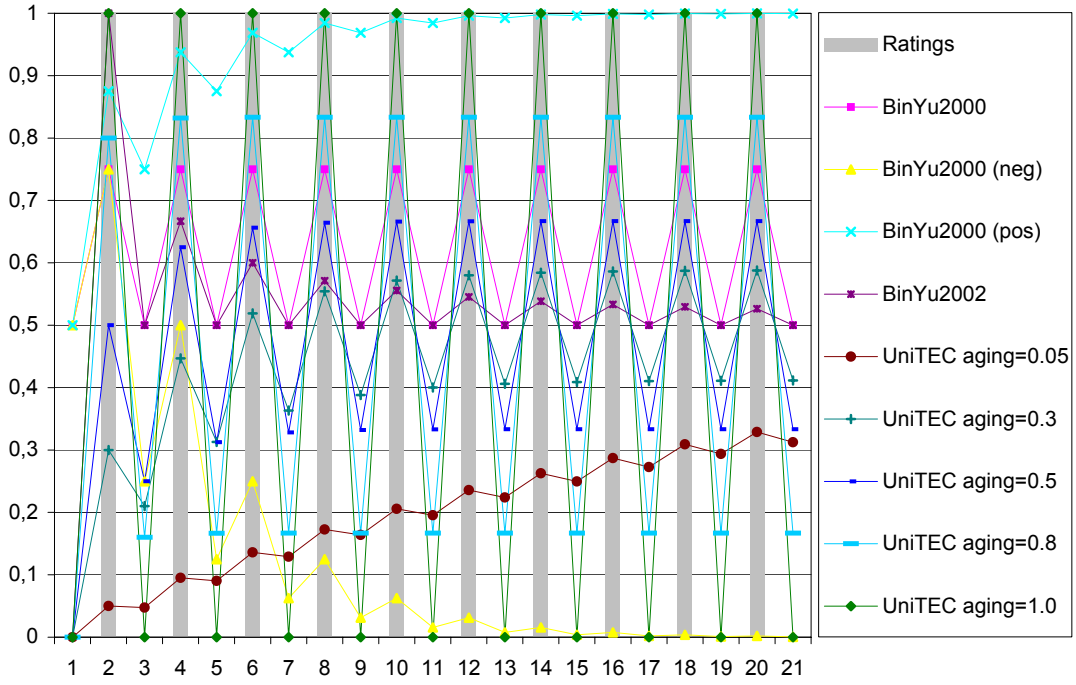


Abbildung B.24.: Vertrauensverlauf im Testszenario *OscilatingRatingsFirstMaximal*, Teil 2

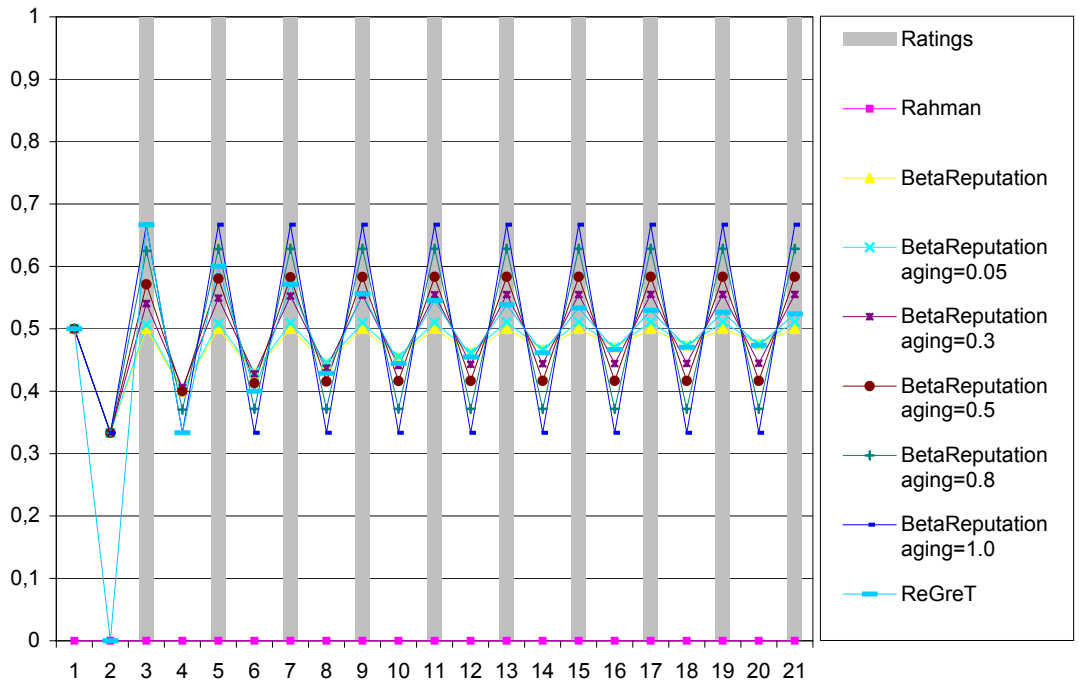


Abbildung B.25.: Vertrauensverlauf im Testszenario *OscilatingRatingsFirstMinimal*, Teil 1

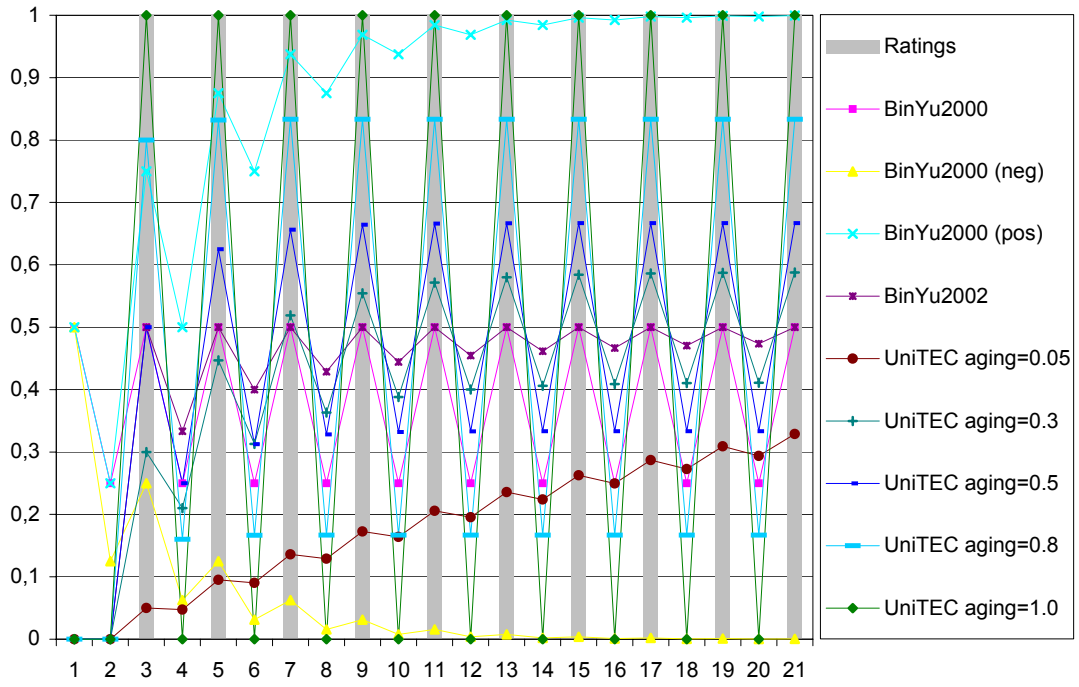


Abbildung B.26.: Vertrauensverlauf im Testszenario *OscilatingRatingsFirstMinimal*, Teil 2

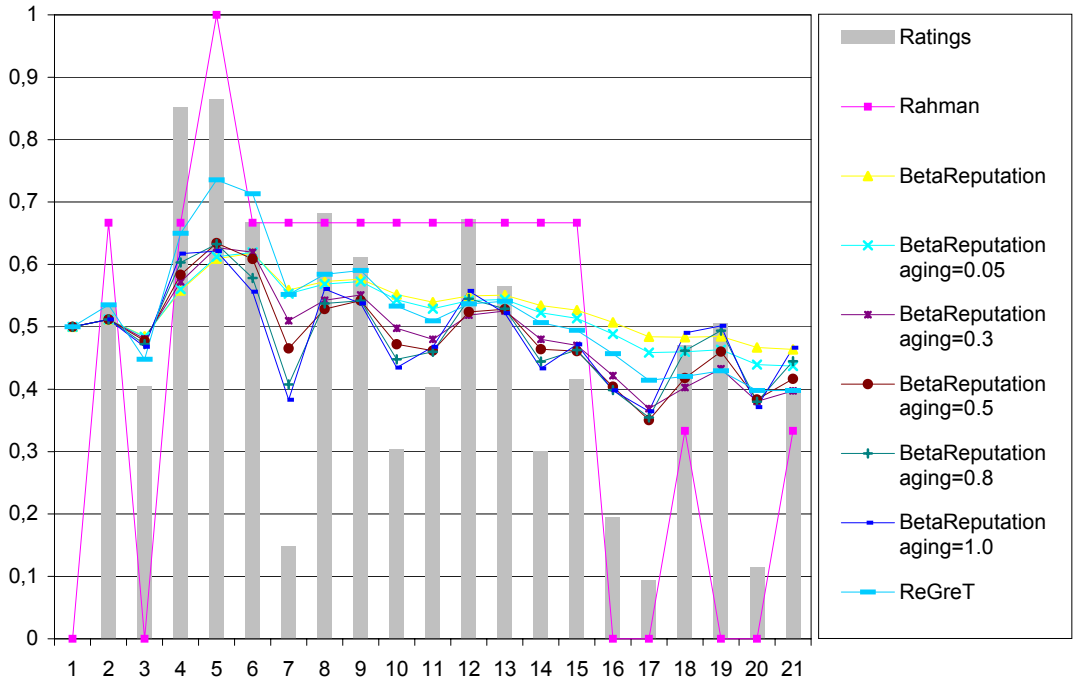


Abbildung B.27.: Vertrauensverlauf im Testszenario *RandomRatings*, Teil 1

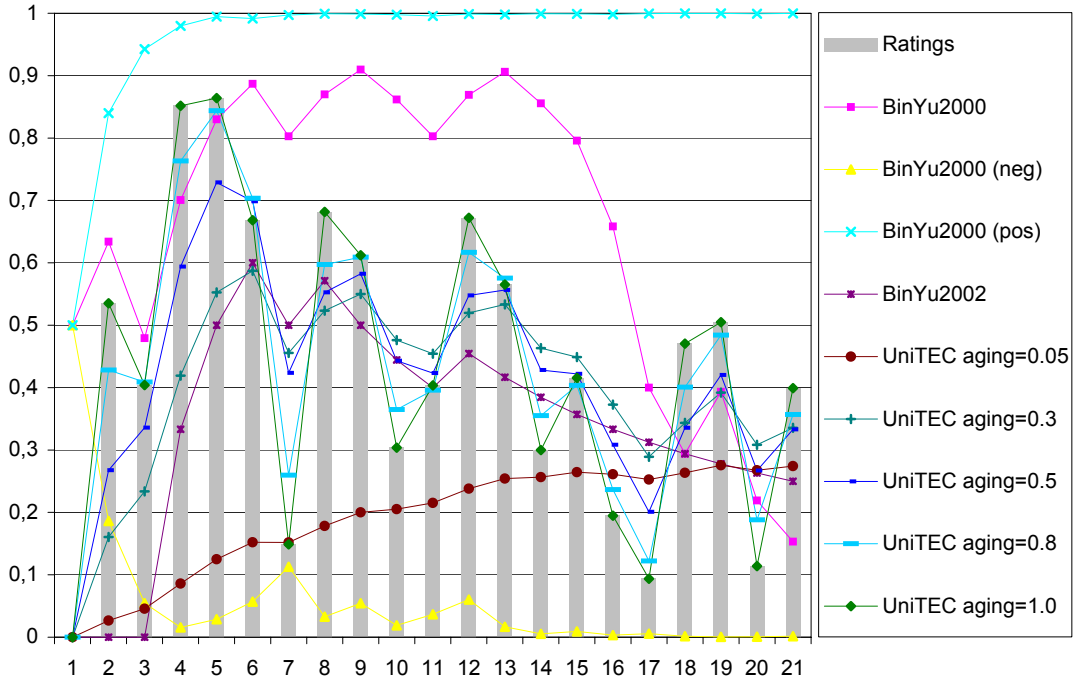
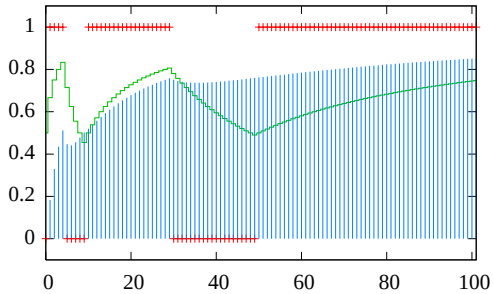
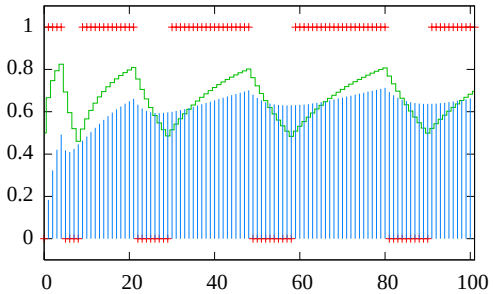


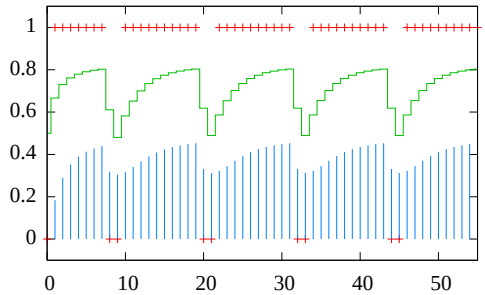
Abbildung B.28.: Vertrauensverlauf im Testszenario *RandomRatings*, Teil 2



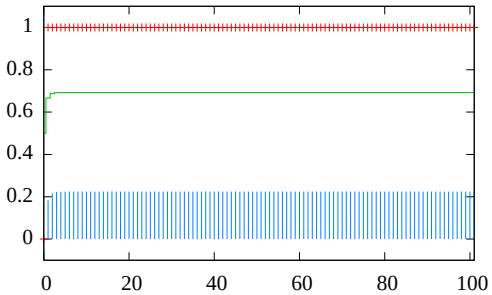
(a) BetaReputation mit Alterungsfaktor $\alpha = 0.0$



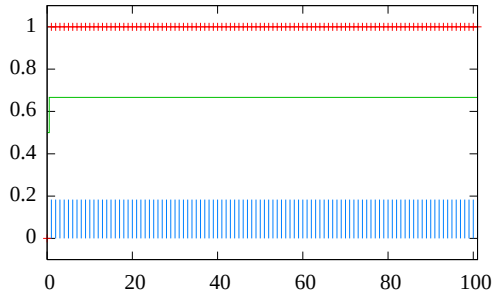
(b) BetaReputation mit Alterungsfaktor $\alpha = 0.05$



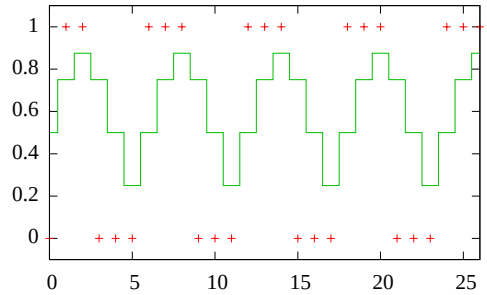
(c) BetaReputation mit Alterungsfaktor $\alpha = 0.3$



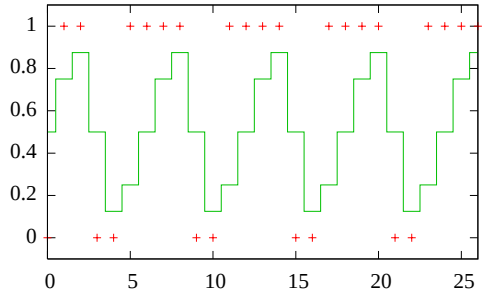
(d) BetaReputation mit Alterungsfaktor $\alpha = 0.8$



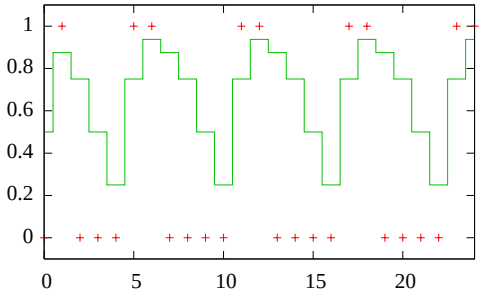
(e) BetaReputation mit Alterungsfaktor $\alpha = 1.0$



(f) Bin Yu (2000)

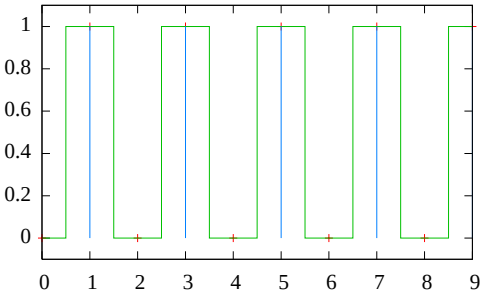


(g) Bin Yu (2000) (neg)

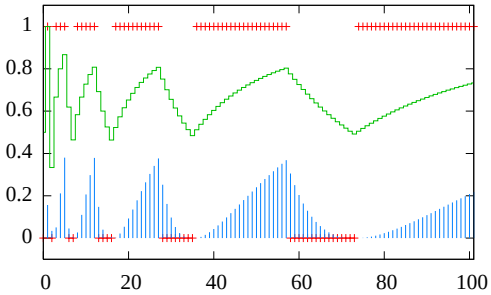


(h) Bin Yu (2000) (pos)

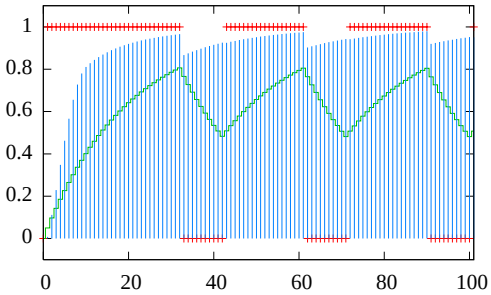
Abbildung B.29.: Vertrauensverlauf im Testszenario *KeepPositive*, Teil 1



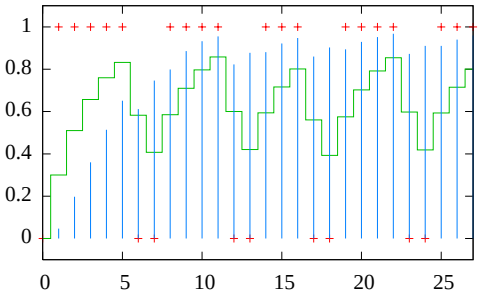
(a) Rahman



(b) ReGreT

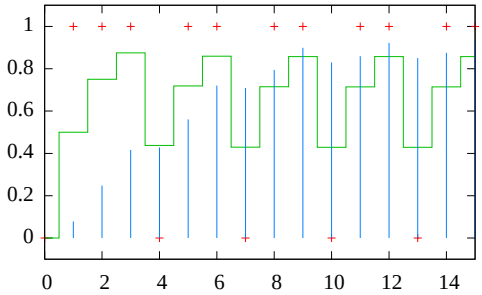


(c) UniTEC mit Alterungsfaktor $\alpha = 0.05$

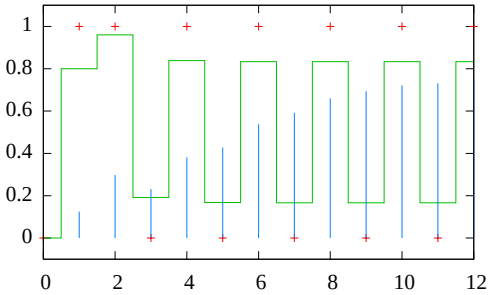


(d) UniTEC mit Alterungsfaktor $\alpha = 0.3$

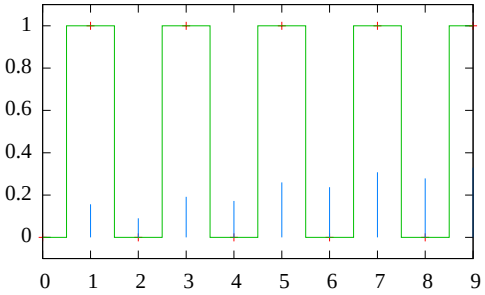
Abbildung B.30.: Vertrauensverlauf im Testszenario *KeepPositive*, Teil 2



(a) UniTEC mit Alterungsfaktor $\alpha = 0.5$



(b) UniTEC mit Alterungsfaktor $\alpha = 0.8$



(c) UniTEC mit Alterungsfaktor $\alpha = 1.0$

Abbildung B.31.: Vertrauensverlauf im Testszenario *KeepPositive*, Teil 3

C. Glossar

UniTEC Allgemein

Agent engl. *agent*
Eine Softwareinstanz, die im Auftrag des Benutzers im UniTEC-System mit anderen Agenten interagiert. Jeder Benutzer meldet sich bei seinem Agenten an, um Aktionen im UniTEC-System durchzuführen (z.B. Anfragen stellen oder Empfehlungen schreiben). Agenten kommunizieren miteinander mit Hilfe eines Peer-to-Peer Netzes.

Benutzer engl. *user*
Die Person, die am UniTEC-System teilnimmt, sei es als Anfragender oder als Empfehler.

Fremdes Pseudonym engl. *foreign pseudonym*
Ein Pseudonym, das bei einem fremden Agenten erstellt wurde und dort einen Benutzer repräsentiert. Im Allgemeinen kennt ein lokaler Benutzer zu einem fremden Pseudonym nur dessen ID und öffentlichen Schlüssel.

Gewissheit engl. *confidence*
Maß der eigenen Sicherheit, dass eine Aussage korrekt ist. Die Gewissheit kann von einer *kompletten Ungewissheit* bis zu einer *totalen Sicherheit* reichen. Im UniTEC-System spielt die Gewissheit z.B. bei den Vertrauensaussagen und bei eigenen Empfehlungen eine Rolle.

Pseudonym engl. *pseudonym*
Anonymer angenommener Name eines Benutzers mit dem er im UniTEC-System teilnimmt, um seine wahre Identität zu verbergen. Ein Benutzer kann ein oder mehrere Pseudonyme erstellen. Bei jeder Teilnahme am System wird dafür eines seiner eigenen Pseudonyme ausgewählt. Mit jedem Pseudonym sind verschiedene Daten assoziiert, wie etwa seine ID oder sein Schlüsselpaar.

UniTEC Universal Trust Architecture for Electronic Commerce
Projekt an der Abteilung „Verteilte Systeme“ des Instituts für Parallele und Verteilte Systeme der Universität Stuttgart, das während einer 3-jährigen Zusammenarbeit mit der Firma Hewlett Packard entstanden ist. Das Projekt beschäftigt sich mit der Frage, wie Vertrauen in einem digitalen System modelliert und weitergegeben werden kann, mit dem langfristigen Ziel, vertrauensbildende Maßnahmen für den Einsatz im e-Commerce-Bereich zu entwickeln.

UniTEC-System
Die Menge aller Komponenten und Entitäten, die mit einem gemeinsamen Ziel mit Hilfe von UniTEC miteinander verknüpft sind.

Empfehlungssystem

Anfragender engl. *query originator*
Ein Benutzer, der nach Empfehlungen zu einem bestimmten Objekt in einer bestimmten Kategorie sucht und dazu eine Anfrage an das UniTEC System stellt.

Anfragewert engl. *query value*
Der Wert (Kosten), der für den Benutzer eine gute Empfehlung für das gesuchte Objekt repräsentiert. Der Benutzer spezifiziert diesen Wert bei seiner Anfrage. Der Vertrauensaktualisierungsalgorithmus kann aufgrund dessen z.B. nach einer Bewertung einer Empfehlung die Vertrauensberechnungen unterschiedlich gewichten: Eine Empfehlung zu einer „kostbaren“ Anfrage hat für den Benutzer eine grössere Bedeutung als eine Empfehlung zu einer „minderwertigen“ Anfrage.

Antwortende engl. *responder*
Jedes Pseudonym, das eine Antwort mit einer Empfehlung an den Anfragenden zurück schickt. Ein Antwortender muss nicht unbedingt ein Pseudonym des Empfehlers sein.

Ausbreitungsalgorithmus engl. *dissemination algorithm*
Beschreibt, auf welche Art eine Anfrage verbreitet wird, d.h. wie die direkte Nachbarschaft für eine konkrete Anfrage berechnet wird.

Benutzeranfrage engl. *user query*
Die Anfrage, die ein Benutzer seinem Agenten zur Verarbeitung im UniTEC-System übergibt.

Direkte Nachbarschaft engl. *direct neighborhood*
Eine Menge von fremden Pseudonymen, die aufgrund einer konkreten Anfrage zur direkten Befragung ausgewählt werden.

Empfehler engl. *recommender*
Ein Benutzer des UniTEC-Systems, der Urheber einer Empfehlung ist.

Empfehlung engl. *recommendation*
Beschreibt die Beurteilung eines Objekts durch einen Benutzer. Diese Beurteilung beruht im Normalfall auf eigener Erfahrung mit diesem Objekt und ist von Natur aus subjektiv. Eine Empfehlung besteht aus beliebigen Mengen an Daten unterschiedlicher Art (textuell, binär, skalar, usw.) zusätzlich zu dem Wert der Gewissheit des Urhebers in die eigene Aussage.

Empfehlungsanfrage engl. *recommendation query*

Eine Nachricht, mit der der Agent eine Benutzeranfrage im Peer-to-Peer Netz mit Hilfe eines Ausbreitungsalgorithmus an seine direkte Nachbarschaft weiterleitet. Eine Empfehlungsanfrage enthält folgende Daten:

- *Pseudonym des Anfragenden*: Damit jeder der die Anfrage antworten will weiss, wohin die Antwort zu senden ist.
- *Eindeutiger ID der Anfrage*: Somit kann eine Anfragende die Zuordnung der eintreffenden Antworten zu den entsprechenden Anfragen machen.
- *Suchanfrage*: Die Anfrage des Benutzers in der UniTEC-Anfragesyntax
- *Kategorie*: Die Kategorie, in der die Anfrage gestellt wurde
- *Hop-Zähler*: Anfänglich durch den Anfragenden spezifiziert, legt dieser Zähler fest, in wie vielen Ebenen in das Nachbarschaftsnetzwerk die Anfrage propagiert werden soll. Vor einer Weiterleitung wird dieser Zähler dekrementiert.
- *Vertrauenskette*: Derjenige, der eine Anfrage vermittelt, ergänzt diese Vertrauenskette mit der Vertrauensaussage, die das Vertrauens in das nachfolgende Pseudonym beschreibt.

Der Agent, der die Anfrage direkt vom Benutzer erhalten hat erstellt die notwendigen Daten aus der Benutzeranfrage. Die Vertrauenskette wird hier initialisiert und enthält ein erstes Element (die Vertrauensaussage in das Pseudonym, an das die Anfrage weitergeleitet wird).

Die Authentizität der Daten werden mittels Signaturen geschützt. Alle Agenten, die die Anfrage weiterleiten, signieren den neuen Eintrag in der Vertrauenskette mit Hilfe des eigenen privaten Schlüssels.

Empfehlungsantwort engl. *recommendation response*

Die Nachrichten mit Empfehlung(en), die ein Agent an den Anfragenden zurückschickt. Eine Antwort kann auch mehrere Empfehlungen zu einer Anfrage enthalten. Diese Nachricht enthält den ID der Anfrage, die signierte Vertrauenskette bis zum Empfehler und die eigentlichen Empfehlungen.

Kategorie engl. *category*

Ein Kontext, in dem Objekte eingeordnet werden können. Jede Anfrage wird gezielt in einer bestimmten Kategorie an das UniTEC-System gestellt.

Kategorien Abhängigkeiten engl. *category dependency*

Eine Kategorie kann gegebenenfalls eine Beziehung zu andere Kategorien haben. Die Stärke dieser Beziehung (semantische Distanz) kann in Form von Gewichten in einem Abhängigkeitsgraphen dargestellt werden. Die Gewichte sind dabei im Wertebereich $[0..1]$, wobei 0 *keine Beziehung* und 1 *uneingeschränkte Beziehung* bedeutet.

Nachbarschaftsnetz engl. *neighborhood net*

Das Netz von Pseudonymen, welches sich aufgrund einer Anfrage bildet. Zunächst wird

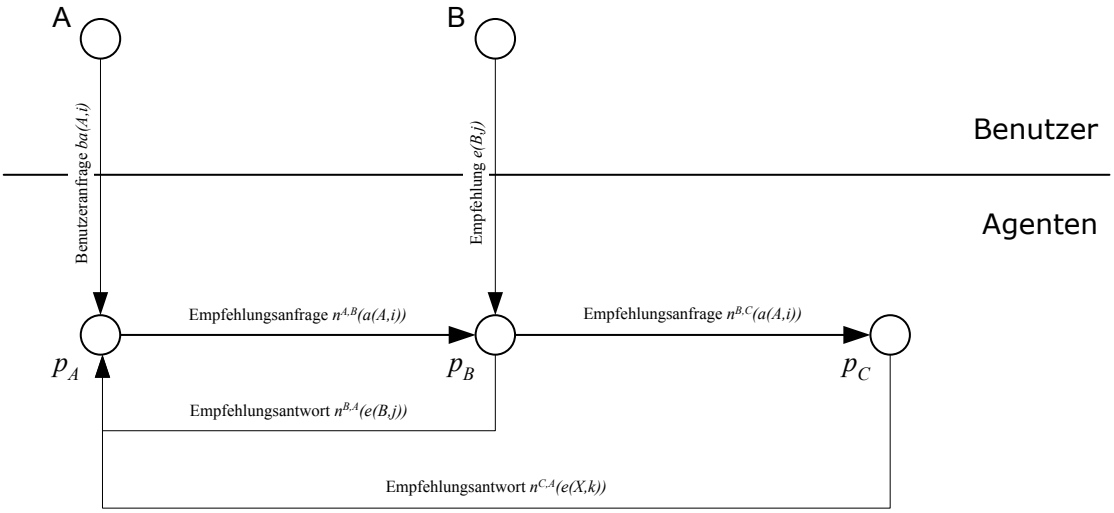


Abbildung C.1.: Beispiel zur Erläuterung einiger Begriffe

in das Netz die direkte Nachbarschaft aufgenommen. Zu jedem dieser Nachbarn kommt dessen Nachbarschaft hinzu. Der Anfragende kann die Anzahl der Ebenen bei der Anfrage definieren (in Form eines Hop-Zählers). Das Nachbarschaftsnetz ist komplett, sobald keine weiteren Nachbarn dazu kommen.

Vermittler engl. *intermediary*
Jedes Pseudonym, das in einer Vertrauenskette zwischen dem Anfragenden und dem Empfehler steht. Somit ist jeder Weiterleiter ein Vermittler, wenn aus dieser Weiterleitung eine Empfehlung an den Anfragenden zurückkommt. Außerdem ist derjenige, der eine Anfrage mit einer Empfehlung beantwortet, aber nicht der Urheber der Empfehlung ist, auch ein Vermittler (aber in diesem Zusammenhang kein Weiterleiter).

Der Anfragende bekommt die Liste aller Vermittler zu jedem Empfehler in jeder Antwort in Form von Vertrauensketten.

Weiterleiter engl. *forwarder*
Jedes Pseudonym, das eine Empfehlungsanfrage an seine direkte Nachbarschaft weiterleitet (mit Ausnahme des Pseudonym des Anfragenden selbst).

Die Beteiligten an einer Empfehlungskommunikation werden in dem Beispiel der Abbildung C.1 deutlich, wobei X ein fremder nicht dargestellter Benutzer ist:

$n^{x,y}$ symbolisiert ein Nachrichtenübermittlung von x an y.

$ba(A,i)$ ist die Benutzeranfrage i von A

$e(B,j)$ ist die Empfehlung j von B

$n^{A,B}(a(A,i))$ ist die Empfehlungsanfrage i von A, die Pseudonym p_A an Pseudonym p_B schickt.

$n^{B,C}(a(A,i))$ ist die Empfehlungsanfrage i von A, die Pseudonym p_B an Pseudonym p_C weiterleitet.

$n^{B,A}(e(B,j))$ ist die Empfehlungsantwort von Pseudonym p_B an Pseudonym p_A mit der Empfehlung j von B (eigene Empfehlung).

$n^{C,A}(e(X,k))$ ist die Empfehlungsantwort von Pseudonym p_C an Pseudonym p_A mit einer fremden Empfehlung k von X.

Damit sind folgende Bezeichnungen für die Rollen in der Kommunikation gültig:

- A ist die *Anfragende*
- B ist ein *Empfehler*
- p_B und p_C sind *Antwortende*
- B und X sind *Empfehler*
- p_B ist der einzige *Weiterleiter*
- p_B und p_C sind *Vermittler* (obwohl p_C kein Weiterleiter ist)

Vertrauensmodell

Abnutzungsfaktor engl. *fading factor*
Gibt einen Maß an, wie schnell der Vertrauenswert im Laufe der Zeit nachlässt, wenn keine neuen Erkenntnisse zu einem Pseudonym in einer Kategorie eintreffen. Die genaue Semantik und Implementierung dieses Faktors ist Aufgabe des Vertrauensaktualisierungsalgorithmus, der eventuell den Abnutzungsfaktor gar nicht berücksichtigt.

Aktualisierungsgewicht engl. *update weight*
Eine Vertrauensaktualisierung kann abhängig von verschiedenen Faktoren unterschiedlich gewichtet werden: z.B. eine Aktualisierung des Vertrauens in einer verwandten Kategorie kann das Gewicht der Aktualisierung um die semantischen Distanz zwischen den Kategorien verringern; oder das Gewicht kann abhängig von Kosten- oder Risikofaktoren anders gewichtet werden. Der Vertrauensaktualisierungsalgorithmus kann diese Gewichtung bei der Berechnung berücksichtigen.

Alterungsfaktor engl. *aging factor*
Gibt einen Maß an, inwiefern neuere Ereignisse im Vergleich zu älteren bei der Berechnung des Vertrauens stärker gewichtet werden sollen. Ein höherer Faktor legt fest, dass schneller auf neue Erkenntnisse reagiert wird. Die genaue Semantik und Implementierung dieses Faktors ist Aufgabe des Vertrauensaktualisierungsalgorithmus, der eventuell den Alterungsfaktor gar nicht berücksichtigt.

Bewertung einer Empfehlung engl. *recommendation rating*
Jede Empfehlung kann durch den Anfragenden bewertet werden (explizit durch den Benutzer oder durch andere implizite Automatismen). Die Bewertung der Empfehlung passiert im Allgemeinen nachdem der Anfragende mit dem Objekt der Anfrage selbst Erfahrungen gemacht hat.

Bewertungshistorie engl. *rating history*
Eine Sammlung von vergangenen Bewertungen eines Benutzers. Zur Berechnung eines Vertrauenswertes kann ein Vertrauensaktualisierungsalgorithmus die vergangenen Bewertungen zu Empfehlungen eines Pseudonyms in der gewünschten Kategorie nutzen. Diese Historie besteht aus einer zeitlich geordneten Menge von Tupeln in der Form *(Zeitstempel, Kategorie, Empfehler, Empfehlergewissheit, Vertrauenskette, Bewertungsgröße, Anfragewert)*:

- *Zeitstempel* ist die Beschreibung des Zeitpunktes, an dem die Bewertung abgegeben wurde.
- *Kategorie* ist die Kategorie in der die Anfrage gestellt wurde, zu der die Empfehlung als Antwort kam.
- *Empfehler* ist das Pseudonym des Empfehlers.
- *Empfehlergewissheit* ist die Gewissheit des Empfehlers in seine Empfehlung.
- *Vertrauenskette* ist die Vertrauenskette vom Anfragendem bis zum Empfehler.
- *Bewertungsgröße* ist das Ergebnis der Bewertung des lokalen Benutzers zu dieser Empfehlung in der Bewerungsmetrik des generischen Vertrauensmodells.
- *Anfragewert* ist der Wert, den der Anfragende für seine Anfrage angegeben hat, auf die sich die Empfehlung, die hier bewertet wird, bezieht.

Bewertungshistorienfenster engl. *rating history window*
Hiermit wird jeweils ein Punkt in der Vergangenheit festgelegt, ab wann die Historie der Bewertungen berücksichtigt wird. Ältere Bewertungen in der Bewertungshistorie sind nicht mehr zugängig (sind somit „vergessen“). Eventuell kann der Benutzer dieses Fenster selbst definieren. Mögliche Varianten dieses Fenster zu beschreiben sind reihenfolgenorientiert (z.B. „die letzten 100 Bewertungen“), zeitorientiert (z.B. „alle Bewertungen der letzten 6 Monate“) oder unbegrenzt. Der Vertrauensaktualisierungsalgorithmus hat nur diese entsprechende Historie zur Berechnung des Vertrauens zur Verfügung.

Bewertungsmetrik engl. *rating value metric*
Eine für den Benutzer einleuchtende Metrik für eine Bewertung. Das generische Vertrauensmodell definiert konkret wie dieser Wert zu realisieren ist. Lokale Vertrauensmodelle können eventuell andere Bewertungsmetriken unterstützen (z.B. skalar im Bereich $[0..1]$, im Schulnotensystem von 1 bis 6 oder mit dem Vergeben von 1 bis 5 „Sternen“), diese müssen jedoch mit Hilfe einer Bewertungsmetrikabbildung in die Bewertungsmetrik des generischen Vertrauensmodells abgebildet werden können.

Bewertungsmetrikabbildung engl. *rating value metric mapping*

Eine bijektive Abbildungsfunktion, die zwei unterschiedliche Bewertungsmetriken ineinander abbildet. Jeder Vertrauensaktualisierungsalgorithmus arbeitet mit einer bestimmten Bewertungsmetrik. Eine Abbildung von und zu der Bewertungsmetrik des generischen Vertrauensmodells ermöglicht erst das Abbarbeiten der Informationen durch den unterschiedlichen Algorithmen.

Erfahrung engl. *experience*

Eine Erfahrung zeigt sich im UniTEC-System in zwei Stellen: Eigene Erfahrung mit Objekten, die zur Formulierung von Empfehlungen genutzt wird und Erfahrungen mit Empfehlungen anderer, auf dessen Basis das Vertrauensverhältnis aufgebaut wird. Die Erfahrungen mit Empfehlungen werden durch den Benutzer konkret in Form von Bewertungen eingegeben, die als Bewertungshistorie die gesammelte Erfahrung des Benutzers mit anderen Teilnehmern darstellt. Die Vertrauensaktualisierungsalgorithmen arbeiten mit diesen Erfahrungen.

Generisches Vertrauensmodell engl. *generic trust model*

Legt ein öffentliches Vertrauensmodell für das UniTEC-System fest. Dieses Vertrauensmodell ist generisch in der Hinsicht, als dass alle weiteren relevanten Vertrauensmodelle darauf abgebildet werden können (mit Hilfe einer Vertrauenswertabbildung). Dieses Modell wird global verwendet und ermöglicht so eine eindeutige Repräsentation von Vertrauen und Bewertungen im UniTEC-System.

Lokales Vertrauensmodell engl. *local trust model*

Vertrauensmodell, das lokal bei einem Agenten integriert ist, um bestimmte Vertrauensaktualisierungsalgorithmen zu ermöglichen, die nicht direkt auf dem generischen Vertrauensmodell arbeiten können. Um im UniTEC-System teilnehmen zu können müssen die Werte des lokalen Vertrauensmodells zurück in die des generischen Vertrauensmodells abgebildet werden.

Reputation engl. *reputation*

Eine Zusammenfassung des Vertrauens aller teilnehmenden Benutzer in einen Pseudonym in einer bestimmten Kategorie und zu einem bestimmten Ziel. Somit ist der Wert der Reputation nicht durch einzelne Benutzer erfassbar oder begreifbar, sondern global. Er kann auch nur geringfügig durch jeden Einzelnen beeinflusst werden. Ein neu dazugekommener Benutzer hat anfänglich keine Reputation, und kann diese aufbauen, indem er z.B. nützliche Empfehlungen schreibt, die von anderen Teilnehmern als „gut“ oder „nützlich“ bewertet werden.

Vertrauen engl. *trust*

Eine subjektive Einschätzung eines Agenten, ob ein anderer Agent in einem bestimmten Kontext zu einen bestimmten Ziel und an einem bestimmten Zeitpunkt mit ihm kooperieren wird. Im Sinne des UniTEC-Systems kann diese Kooperation z.B. das Ziel haben, eine Empfehlung zu einem Objekt abzugeben, die mit der Meinung des Anfragenden zu dem Objekt übereinstimmt. Vertrauen hat eine subjektive Bedeutung, wird lokal berech-

net (durch jedem Benutzer selbstständig), ist Pseudonym-, Kontextspezifisch (Kategoriespezifisch), asymmetrisch, hat einen Ziel und kann sich im Laufe der Zeit verändern.

Vermittlungsvertrauen engl. *intermediation trust*
Vertrauen in ein fremdes Pseudonym, mit dem bestimmten Ziel, dass es eine Anfrage in einer bestimmten Kategorie vermitteln wird. Diese Art von Vertrauen wird auch *indirektes Vertrauen* genannt.

Empfehlungsvertrauen engl. *recommendation trust*
Vertrauen in ein fremdes Pseudonym, mit dem bestimmten Ziel, dass es auf eine Anfrage in einer bestimmten Kategorie mit einer nützlichen Empfehlung antworten wird. Diese Art von Vertrauen wird auch *direktes Vertrauen* genannt.

Vertrauensaktualisierung engl. *trust update*
Die Veränderung einer oder mehrerer Vertrauensaussagen zu bestimmten Pseudonymen, die durch ein Ereignis (z.B. der Eingang einer neuen Bewertung) ausgelöst wird.

Vertrauensaktualisierungsalgorithmen engl. *trust update algorithm*
Der Algorithmus, der eine Vertrauensaktualisierung durchführt. Möglicherweise kann ein Vertrauensaktualisierungsalgorithmus mit dem generischen Vertrauensmodell nicht arbeiten und verwendet deswegen ein lokales Vertrauensmodell als Adapter. Er ändert möglicherweise den Zustand des Vertrauensmodells und kann aus unterschiedlichen Komponenten Informationen zur Berechnung hinzuziehen (Bewertungshistorie, Reputation über eine Reputationsanfrage, benutzerspezifische Parameter, usw.).

Vertrauensbeziehung engl. *trust relation*
Oft auch Vertrauensaussage. Ein Tupel der Form $(p_1, p_2, k, z, T_{p_1,p_2}(k,z))$. Hierbei ist $T_{p_1,p_2}(k,z)$ der Vertrauenswert, der das Vertrauen von Pseudonym p_1 in p_2 für Anfragen in der Kategorie k zu dem Ziel z beschreibt.

Vertrauenskette engl. *trust chain*
Eine geordnete Menge von Vertrauensaussagen. Die Kette stellt eine transitive Beziehung dar und ergibt sich aus:

$$(p_1, p_2, T_{p_1,p_2}(k,z_w)), \ldots, (p_{n-2}, p_{n-1}, T_{p_{n-2},p_{n-1}}(k,z_w)), (p_{n-1}, p_n, T_{p_{n-1},p_n}(k,z_e))$$

Im ersten Tupel der Kette ist p_1 der Anfragende. In einer Antwortnachricht ist p_n im letzten Tupel das Pseudonym des Urhebers der Empfehlung, so dass transitiv das Vertrauen des Anfragenden in den Empfehler berechnet werden könnte, falls das Vertrauensmodell dies vorsieht. z_w beschreibt den Zweck der Weiterleitung und z_e den der Empfehlung.

Vertrauensmodell engl. *trust model*
Definiert eine konkrete Realisierung von Vertrauenswerten, die Bewertungsmetrik und beschreibt den Zusammenhang des Vertrauens mit anderen Komponenten im System (z.B. Kategorien, Pseudonyme, Bewertungshistorie).

Vertrauenswert engl. *trust value*

Wert, durch den das Vertrauensverhältnis zwischen einem Benutzer und anderen Pseudonymen ausgedrückt werden kann. Das Vertrauensmodell definiert, wie der Vertrauenswert implementiert wird (z.B. als skalarer Wert). Die Werte stellen Einschätzungen zwischen *komplettes Misstrauen*, *kein Vertrauen* (wenn noch keine Erfahrungen gemacht wurden) und *volles (blindes) Vertrauen* dar.

Vertrauenswertabbildung engl. *trust value mapping*

Die Abbildung zwischen Vertrauenswerte zweier unterschiedlichen Vertrauensmodelle. Somit können die Vertrauenswerte des generischen Vertrauensmodells für bestimmte Vertrauensaktualisierungsalgorithmen in andere Darstellungen umgewandelt werden.

Vertrauenswertinitialisierung engl. *trust value initialisation*

Ein Algorithmus, der einen neuen fremden Pseudonym, zu dem noch kein Vertrauenswert bekannt ist, einen Vertrauenswert zuordnet. Die Zuordnung kann z.B. den Vertrauenswert auf „kein Vertrauen“ setzten, oder über eine Reputationsanfrage einen initialen Wert im Nachbarschaftsnetz finden.

D. Literaturverzeichnis

[Abdul-Rahman und Hailes 2000] ABDUL-RAHMAN, Alfarez ; HAILES, Stephen: Supporting Trust in Virtual Communities. In: *Proceedings Hawaii International Conference on System Sciences 33, Maui, Hawaii*, 4-7 January 2000

[Aberer und Despotovic 2001] ABERER, K. ; DESPOTOVIC, Z.: Managing Trust in a Peer-2-Peer Information System. In: *Proceedings of the Ninth International Conference on Information and Knowledge Management (CIKM 2001), Atlanta*, 2001

[Castelfranchi und Falcone 2000] CASTELFRANCHI, Cristiano ; FALCONE, Rino: Trust Is Much More than Subjective Probability: Mental Components and Sources of Trust. In: *Proceedings of the 33rd Hawaii International Conference on System Sciences, Maui, Hawaii* Bd. 6 National Research Council - Institute of Psychology (Veranst.), January 2000, S. 6008

[Dellarocas 2003a] DELLAROCAS, Chrysanthos: Building Trust Online: The Design of Robust Reputation Reporting Mechanisms in Online Trading Communties. (2003), September

[Dellarocas 2003b] DELLAROCAS, Chrysanthos: The Digitization of Word-of-Mouth: Promise and Challenges of Online Feedback Mechanisms. In: *Management Science* (2003), October

[Gambetta 2000] GAMBETTA, Diego: *Can We Trust Trust?* S. 213–237. In: *Trust: Making and Breaking Cooperative Relations, electronic edition, Chapter 13*, Department of Sociology, University of Oxford, 2000

[Gnuplot 2004] GNUPLOT: *Gnuplot*. http://www.gnuplot.info/. September 2004

[Huberman und Wu 2003] HUBERMAN, Bernardo A. ; WU, Fang: The Dynamics of Reputations. In: *Computing in Economics and Finance* 18 (2003)

[Jonker und Treur 1999] JONKER, Catholijn M. ; TREUR, Jan: Formal Analysis of Models for the Dynamics of Trust based on Experiences. In: *Agent System Engineering, Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW'99*, 1999

[Jøsang u. a. 2003a] JØSANG, Audun ; GRAY, Elizabeth ; KINATEDER, Michael: Analysing Topologies of Transitive Trust. In: *Proceedings of the Workshop of Formal Aspects of Security and Trust (FAST), Pisa, Italy*, September 2003

[Jøsang u. a. 2003b] JØSANG, Audun ; HIRD, Shane ; FACCER, Eric: Simulating the Effect of Reputation Systems on e-Markets. In: *Proceedings of the First International Conference on Trust Management, Crete*, May 2003

[Jøsang und Ismail 2002] JØSANG, Audun ; ISMAIL, R.: The Beta Reputation System. In: *Proceedings of the 15th Bled Conference on Electronic Commerce, Bled, Slovenia*, 17-19 June 2002

[Jøsang und McAnally 2004] JØSANG, Audun ; MCANALLY, David: Multiplication and Comultiplication of Beliefs. In: *Submitted to the International Journal of Approximate Reasoning* (2004)

[Kinateder und Rothermel 2003] KINATEDER, Michael ; ROTHERMEL, Kurt: Architecture and Algorithms for a Distributed Reputation System. In: *Proc. of the First International Conference on Trust Management, no. 2692 in LNCS, iTrust, Crete, Greece*, Springer-Verlag, 2003, S. 1–16

[Macy und Skvoretz 1996] MACY, Michael W. ; SKVORETZ, John: The Evolution of Trust and Cooperation between Strangers: A Computational Model. In: *American Sociological Review* (1996), October

[Marsh 1994] MARSH, Stephen P.: *Formalising Trust as a Computational Concept*, Department of Mathematics and Computer Science, University of Stirling, Dissertation, 1994

[McKnight und Chervany 1996] MCKNIGHT, D. H. ; CHERVANY, Norman L.: The Meanings of Trust / University of Minnesota, Management Information Systems Reseach Center. 1996. – Technical Report

[Mui 2003] MUI, Lik: *Computational Models of Trust and Reputation: Agents, Evolutionary Games, and Social Networks*, Massachusetts Institute of Technology, Dissertation, 2003

[Mui u. a. 2001] MUI, Lik ; MOHTASHEMI, Mojdeh ; ANG, Cheewee ; SZOLOVITS, Peter ; HALBERSTADT, Ari: Ratings in Distributed Systems: A Bayesian Approach / MIT Laboratory for Computer Science. May 2001. – Technical Memo

[Sabater 2003] SABATER, Jordi: *Trust and Reputation for Agent Societies*, Institut d'Investigació en Intelligència Articial, Bellaterra, Dissertation, 2003

[Sabater und Sierra 2000] SABATER, Jordi ; SIERRA, Carles: REGRET: A reputation model for gregarious societies / IIIA. October 2000. – Research Report

[Sabater und Sierra 2002] SABATER, Jordi ; SIERRA, Carles: Reputation and social network analysis in multi-agent systems. In: *First International Conference on Autonomous Agents and Multiagent systems (AAMAS-02), Bologna, Italy*, 2002, S. 475–482

[Terdic 2003] TERDIC, Ralf: Konzeption und Realisierung eines Peer-to-Peer Empfehlungsdienstes / Universität Stuttgart. 2003. – Diplomarbeit

[Weisstein 2004a] WEISSTEIN, Eric W.: *Beta Function - from MathWorld*. http://mathworld.wolfram.com/BetaFunction.html. September 2004

[Weisstein 2004b] WEISSTEIN, Eric W.: *Gamma Function - from MathWorld*. http://mathworld.wolfram.com/GammaFunction.html. September 2004

[Westermann 2004] WESTERMANN, Achim: *JChart2D - Precise Visualisation of Data*. http://jchart2d.sf.net/. September 2004

[White u. a. 2004] WHITE, Scott ; O'MADADHAIN, Joshua ; FISHER, Danyel ; BOEY, Yan-Biao: *Java Universal Network/Graph Framework*. http://jung.sourceforge.net/. June 2004

[Widmaier 2004] WIDMAIER, Jochen: Identitäts-Management in Peer-2-Peer-Umgebungen / Universität Stuttgart. 2004. – Diplomarbeit

[Winkler 1972] WINKLER, Robert L.: *Introduction to Bayesian Inference and Decision*. Holt, Rinehart and Winston, Inc., 1972. – ISBN 0-03-081327-1

[Yu und Singh 2000] YU, Bin ; SINGH, Munindar P.: A Social Mechanism of Reputation Management in Electronic Communities. In: KLUSCH, Matthias (Hrsg.) ; KERSCHBERG, Larry (Hrsg.): *Proceedings of the 4th International Workshop on Cooperative Information Agents*, 2000, S. 154–165

[Yu und Singh 2002] YU, Bin ; SINGH, Munindar P.: An evidential model of distributed reputation management. In: *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems*, July 2002, S. 294–301

[Ziegler und Lausen 2004] ZIEGLER, Cai-Nicolas ; LAUSEN, Georg: Analyzing Correlation Between Trust and User Similarity in Online Communities. In: *2nd International Conference on Trust Management (iTrust 2004), Oxford, UK*, 29-31 March 2004

Stichwortverzeichnis

Abnutzung, 28
    Ergebnisse der Umsetzung, 84
    Faktor, 141
    im generischem Vertrauensmodell, 44
    Umsetzung, 68
Agent, 137
Aktualisierungsgewicht, 141
Aktualisierungstyp, 55
Alterung
    Faktor, 141
    im generischem Vertrauensmodell, 43
Alterungsfaktor, 28
    Umsetzung, 65
Anfragender, 138
Anfragewert, 138
Antwortende, 138
Ausbreitungsalgorithmus, 138

Benutzer, 137
Benutzeranfrage, 138
Beta Reputation System, 35
    Analyse der Testergebnisse, 77
    Metrikskalierung, 61
    Umsetzung in UniTEC, 59
Bewertung, 142
    Abbildung der Metrik, 143
    Metrik, 142
Bewertungen, 26
    Historienfenster, 28
Bewertungshistorie, 142
Bewertungshistorienfenster, 28, 142
    Umsetzung, 67
Bewertungsmetrik
    Generisches Vertrauensmodell, 42
Bin Yu [2000], 33
    Analyse der Testergebnisse, 71
    Umsetzung in UniTEC, 56
    Vertrauensmodell, 33
Bin Yu [2002], 34
    Analyse der Testergebnisse, 76
    Umsetzung in UniTEC, 58
    Vertrauensmodell, 34

Empfehler, 138
Empfehlung, 138
    Bewertung, 142
Empfehlungsanfrage, 139
Empfehlungsantwort, 139
Empfehlungsvertrauen, 144
Ereignisse
    NewRating, 107
    TrustValueChange, 104
Erfahrung, 143
Erfahrungen, 26

Generisches Vertrauensmodell, 143
Gewichtung
    absolute, 66
    einer Aktualisierung, 141
    im generischem Vertrauensmodell, 46
    relative, 66
    Umsetzung, 66
Gewissheit, 137
    Erwartetes Verhalten der, 30
    Metrik im generischen Vertrauensmodell, 42
    Vertrauens-, 24

Jonker, 32
    Analyse der Testergebnisse, 83

Umsetzung in UniTEC, 56
Vertrauensmodell, 32

Kategorie, 139
Semantische Verwandtschaft, 139
Klassen
Category, 95
CategoryManagerFactory, 97
DisplayGraphHandler, 118
NewRatingEvent, 107
ProcessTestScenarioAsGraph, 118
ProcessTestScenarios, 118
Rating, 113
RatingsManager, 115
RatingValue, 113
SaveCsvfilesHandler, 117
SaveGnuplotHandler, 117
TrustAlgorithm, 109
TrustIdentifier, 99
TrustManager, 102
TrustStorageEntry, 112
TrustValue, 100
Kontexte, 25

Lik Mui, 37
Analyse der Testergebnisse, 83
Vertrauensmodell, 37
Lokales Vertrauensmodell, 143

Nachbarschaft, 138
Nachbarschaftsnetz, 139

Pseudonym, 137
Fremdes, 137

Rahman, 34
Analyse der Testergebnisse, 73
Umsetzung in UniTEC, 57
Vertrauensmodell, 34
ReGreT, 36
Analyse der Testergebnisse, 81
Umsetzung in UniTEC, 62
Vertrauensmodell, 36
Reputation, 143
Definition, 17

Schnittstellen
CategoryAccessor, 96
CategoryManager, 96
TestScenarioOutputListener, 117
TrustAlgorithmManagerInterface, 104
TrustClientInterface, 103
TrustGatherer, 108
TrustInitialiser, 106
TrustTestScenario, 116
TrustUpdateEventListener, 107
TrustValueChangeListener, 104
Skalare Metriken, 55

Testszenarien, 70
TrustUpdateEvent-Klasse, 106

UniTEC, 137
Abnutzungsfaktor, 84
Analyse der Testergebnisse, 83
Anwendungsfälle, 47
Architektur der Vertrauenskomponente, 49
Use Cases, 47
Vertrauensmodell, 38

Vermittler, 140
Vermittlungsvertrauen, 144
Vertrauen, 143
Definition, 15
Erwartetes Verhalten der Metrik für, 30
Initialisierung, 145
Initialisierung von, 29, 64
Metrik im generischen Vertrauensmodell, 41
Vertrauensaktualisierung, 144
Vertrauensaktualisierungsalgorithmus, 29, 144
Vertrauensaktualisierungsereignis, 24
Gewichtung, 25
Vertrauensbeziehung, 144
Vertrauensgewissheit, 24
Vertrauensinitialisierung, 29
Vertrauenskette, 144
Vertrauensketten, 54
Vertrauensmetrik, 23
Vertrauensmodell, 144

generisches, 143
lokales, 143
Vertrauensverhältnis
Definition, 16
Vertrauenswert, 145
Vertrauenswertabbildung, 145
Vertrauensziel, 16, 26

Weiterleiter, 140

Erklärung

Hiermit versichere ich, diese Arbeit selbstständig verfasst und nur die angegebenen Quellen benutzt zu haben.

_____

(Ernesto Baschny)

Stuttgart, den 15. Oktober 2004