UNIVERSITÄT
KOBLENZ · LANDAU
Institut für Wirtschafts-
und Verwaltungsinformatik

FB 4
Informatik

IT-Sicherheitsmodelle

Rüdiger Grimm

Nr. 3/2008

Arbeitsberichte aus dem Fachbereich Informatik

Arbeitsberichte des Fachbereichs Informatik

ISSN (Print): 1864-0346
ISSN (Online): 1864-0850

Herausgeber / Edited by:

Der Dekan:
Prof. Dr. Zöbel

Kontaktdaten der Verfasser

Rüdiger Grimm
Institut für Wirtschafts- und Verwaltungsinformatik
Fachbereich Informatik
Universität Koblenz-Landau
Universitätsstraße 1
D-56070 Koblenz
EMail: grimm@uni-koblenz.de

IT-Sicherheitsmodelle

Rüdiger Grimm, 23.2.2008

In gekürzter Form, mit Stichworten und Kontrollfragen versehen, veröffentlicht in: WISU – das Wirtschaftsstudium. (Herausgeber: Hartmann-Wendels, Thome, Woll), http://www.wisu.de, vorauss. 37 (2008) 5.

Inhaltsübersicht:

Zusammenfassung

Es wird erklärt, was ein Beschreibungsmodell ist, wie IT-Sicherheitsmodelle grundsätzlich aufgebaut sind und welchem Zweck sie dienen. Zur Illustration werden vier verschiedene IT-Sicherheitsmodelle vorgestellt, die historisch zu unterschiedlichen Zeiten entstanden sind. Sie passen zu vier verschiedenen typischen Anwendungsumgebungen und charakterisieren die zugehörigen Sicherheitsanforderungen. Vorgestellt werden das Bell-LaPadula-Modell zum Vertraulichkeitsschutz in hierarchischen Institutionen, das Clark-Wilson-Modell zum Integritätsschutz von Geschäftsanwendungen, das Chinese-Wall-Modell zum Konkurrentenschutz von Unternehmen und das Gleichgewichtsmodell zum Schutz verbindlichen Handelns im offenen Internet.

1. Einführung: Wozu IT-Sicherheitsmodelle?

Modelle sind ganz allgemein sprachliche oder bildnerische Darstellungen von Objekten der Wirklichkeit mit dem Ziel, gewisse wichtige Aspekte der Objekte hervorzuheben und in ihrem Anwendungskontext zu erklären. Das Hervorheben wichtiger Aspekte und das Weglassen unwichtiger Aspekte nennt man Abstraktion. Modelle sind demnach abstrakte Beschreibungen oder Darstellungen ihrer Gegenstände. Beispiele für Modelle sind

- Architekturmodelle wie Nachbildungen von Gebäuden im Kleinen,
- physikalische Anschauungsmodelle wie das Atommodell der Korpuskel-Wellendualität oder das Urknallmodell des Weltalls,
- chemische Modelle wie der Kohlenstoffring,
- biologische Modelle wie die DNA-Spirale,
- geographische Modelle wie Landkarten und Globen.

Alle diese Modelle haben gemeinsam, dass sie dem Betrachter das beschriebene Objekt vereinfacht ersetzen und für die Erklärung der hervorgehobenen wichtigen Aspekte benutzt werden können. Das Pappmodell eines geplanten Eigenheims etwa ignoriert zum Beispiel das Material, die Stabilität oder die Größe des Originals und hebt statt dessen Aspekte wie Größenverhältnisse, die Lage von Fenstern und Türen und die visuelle Einbettung in die Garten- und Straßenumgebung hervor.

Für die Informationstechnik dienen Modelle als Baupläne oder zur Anschauung von Funktionalität. Beispielsweise beschreibt das Von-Neumann-Modell einer Rechnerarchitektur die gleichartige Behandlung von Verarbeitungsdaten, Adressen und Steuerinformationen im Speicherraum (von Neumann 1945). Damit kann man beispielsweise die Gefahr der Fehlinterpretation eines Verarbeitungsdatums als Steuerinformation durch einen Speicherüberlauf-Angriff erklären.

In der Theorie der IT-Sicherheit beschreiben Modelle Systemzustände und ihre Übergänge, unterscheiden sichere von unsicheren Zuständen und erklären, unter welchen Umständen sichere Zustände erreicht werden. Bei dem eben erwähnten Speicherüberlauf beispielsweise wird dadurch ein unsicherer Zustand erreicht, dass ein Systembefehl, der als passives Anwendungsdatum keine Wirkung erzielen würde, unbefugt in einen Speicherbereich hineingeschrieben wird, aus dem der Prozessor seine Befehle bezieht, und auf diese Weise Schaden anrichten kann. Ein zugehöriges Sicherheitsmodell würde solche Systemübergänge kennzeichnen und Umstände beschreiben, unter denen sie vermieden werden.

Insbesondere dienen IT-Sicherheitsmodelle als Baupläne für sichere Systeme oder Systemkomponenten. In dieser Rolle zeigen sie Sicherheitsanforderungen auf. Produkte, die daraufhin erfolgreich geprüft worden sind, diese Sicherheitsanforderungen zu erfüllen, können von Anwendern, die ja in der Regel keine Informatiker sind, ruhigen Gewissens eingesetzt werden. Für den Einsatz in manchen sensiblen Anwendungen sind solche Sicherheitsprüfungen sogar gesetzlich vorgeschrieben, zum Beispiel für die Erzeugung und Verifikation qualifizierter elektronischer Signaturen. Hier sollen sich Erzeuger und Empfänger signierter Nachrichten auf ihre Geräte verlassen können, dass sie korrekt arbeiten und insbesondere zuverlässig anzeigen, ob eine elektronische Unterschrift echt oder gefälscht ist. Darauf würde sich bei einem sicherheitsgeprüften Signaturgerät im Streitfall sogar ein Richter verlassen.

Einen internationalen Standard zur Formulierung und Überprüfung von IT-Sicherheitsanforderungen bilden die so genannten „Common Criteria“ (CC 2006). Jede nach den Regeln der "Common Criteria" formulierte Menge von Anforderungen folgt einem

Sicherheitsmodell, das ihre Konsistenz und Zielführung plausibel macht. Ein Modell kann sogar – ganz im Sinne von Bauplänen – Hinweise darauf enthalten, welche Art Sicherheitsmechanismen geeignet sind, die Anforderungen zu erfüllen. Das Modell kann mehr oder weniger formal sein, aber je höher die Vertrauenswürdigkeitsstufe eines Prüfgegenstands ist, desto präziser, desto formaler muss das Sicherheitsmodell ausgeführt sein. Es gibt also ein sehr konkretes Motiv, IT-Sicherheitsmodelle zu entwerfen und sie bis zur formalen Genauigkeit hin auszuarbeiten.

Nach dem Glossar des Kriterienkatalogs "Common Criteria" (CC 2006) ist ein Modell "formal", wenn es in einer wohldefinierten Syntax formuliert ist, die mathematische Ableitungsregeln zulässt ("expressed in a restricted syntax language with defined semantics based on well-established mathematical concepts", CC 2006, Part 1: Introduction and General Model, Terms and Definitions). Halbformale oder informelle Modelle müssen ihren Gegenstand in klar umrissenen Begriffen jedenfalls immer noch so genau beschreiben, dass aus den Grundeigenschaften überzeugende Schlüsse auf ein modellgetreues Verhalten gezogen werden können. Für IT-Sicherheitsmodelle heißt „modellgetreues Verhalten" das Verbleiben in sicheren Systemzuständen. Was aber ein sicherer Systemzustand ist und wie sichere Zustände erreicht werden können, genau das muss von einem IT-Sicherheitsmodell beschrieben werden.

2. Allgemeine Form von IT-Sicherheitsmodellen

Sicherheit beruht immer auf dem Ausgleich von Interessenskonflikten von Menschen (Grimm 1994) und hängt daher in hohem Maße von dem jeweiligen Anwendungszusammenhang ab. Zum Beispiel folgt die Sicherheit eines Geldauszahlungsautomaten ganz anderen Regeln als die Sicherheit einer Signalanlage, einer Fließbandsteuerung, eines E-Mail-Servers oder einer Personalaktenablage. Manche Sicherheitsziele sind geradezu entgegengesetzt. Zum Beispiel muss die Aufbewahrung von elektronischen Stimmzetteln zwingend anonym erfolgen, dagegen müssen Vertragsunterschriften unabstreitbar auf ihre Urheber verweisen. Signalanlagen müssen neben einer garantierten Funktionstreue vor allem in ihrer Integrität geschützt werden, während ihre Vertraulichkeit keine Rolle spielt, bei Personalakten und Geldauszahlung dagegen ist gerade die Vertraulichkeit ein wichtiges Gebot. Daher ist es kein Wunder, dass es kein allgemeines „Obersicherheitsmodell" für alle Anwendungen gibt. Vielmehr verlangt jede Anwendungsumgebung ein eigenes IT-Sicherheitsmodell. Deshalb muss jedes IT-Sicherheitsmodell an erster Stelle sein „übergeordnetes Sicherheitsziel" festlegen, das die spezifischen Sicherheitsanforderungen einer Anwendung charakterisiert.

Alle IT-Sicherheitsmodelle enthalten die folgenden *fünf Beschreibungselemente*:

1. die Definition eines übergeordneten Sicherheitsziels,
2. die Spezifikation sicherer Systemzustände,
3. ein Regelwerk für erlaubte Zustandsübergänge und
4. ein Sicherheitstheorem, in dem bewiesen wird, dass bei Einhaltung des Regelwerks das System von sicheren Zuständen notwendig in sichere Zustände überführt wird.
5. Ein Vertrauensmodell beschreibt, mit welcher Systemauslegung und unter welchen Annahmen über die Anwendungsumgebung die sicheren Systemzustände das Sicherheitsziel erreichen.

Ein Modell hat *drei Lücken* zu schließen, nämlich die Lücken

1. zwischen dem Sicherheitsziel und seinen Anwendungen,
2. zwischen den sicheren Systemzuständen und dem Sicherheitsziel
3. und zwischen den erlaubten Zustandsübergängen und den sicheren Systemzuständen.

Zur ersten Lücke: Um zu prüfen, ob eine konkrete Anwendung sich an einem bestimmten IT-Sicherheitsmodell orientieren kann, prüft man zunächst, ob die Definition des übergeordneten Sicherheitsziels für diese Anwendung zutrifft. Wenn das der Fall ist, würde man für diese Anwendung nur solche Produkte einsetzen, die zu den Spezifikationen des Modells konform sind. Um ganz sicher zu gehen, würde man verlangen, dass die einzusetzenden Produkte auf Übereinstimmung mit dem Modell geprüft und zertifiziert sind. Das ist die Aufgabe des Anwenders.

Die zweite Lücke schließt das Vertrauensmodell (fünftes Beschreibungselement eines Sicherheitsmodells). Es enthält Annahmen über die Implementierung eines Systems und über das Verhalten der Anwendungsumgebung, unter denen dieses System in dieser Umgebung das übergeordnete Sicherheitsziel dadurch erreicht, dass es in den sicheren Zuständen verbleibt. Das Vertrauensmodell ist nicht vollständig formalisierbar, sondern Gegenstand eines argumentativen Diskurses von Sicherheits- und Anwendungsexperten.

Die dritte Lücke schließt das Sicherheitstheorem (viertes Beschreibungselement eines Sicherheitsmodells) mit seinem Nachweis, dass erlaubte Zustandsübergänge einen sicheren Systemzustand wieder in einen sicheren Systemzustand überführen. Die Systemzustände und die erlaubten Zustandsübergänge müssen explizit und möglichst genau beschrieben sein. Die sicheren Systemzustände können zunächst umgangssprachlich als spezielle Sicherheitsziele (im Sinne der CC 2006) formuliert sein. Bei entsprechend hohen Anforderungen müssen die sicheren Systemzustände sogar mathematisch formuliert und das zugehörige Sicherheitstheorem formal bewiesen werden. Bei etwas geringeren Anforderungen würde eine sprachlich überzeugende Formulierung des Modells genügen.

In den folgenden Abschnitten werden vier IT-Sicherheitsmodelle skizziert, die für vier charakteristische Sicherheitsziele stehen, und zwar

- in Abschnitt 3 das Bell-LaPadula-Modell (Bell/LaPadula 1973 und 1975) mit dem Sicherheitsziel der Vertraulichkeit in gemeinsam genutzten Ressourcen: vertrauliche Informationen gelangen nicht in die Hand unautorisierter Personen;
- in Abschnitt 4 das Clark-Wilson-Modell (Clark/Wilson 1987 und 1988) mit dem Sicherheitsziel der Integrität der Daten: das Anwendungssystem repräsentiert seine Anwendung korrekt;
- in Abschnitt 5 das Chinese-Wall-Modell (Brewer/ Nash 1989) mit dem Sicherheitsziel der Verhinderung von Insiderhandel: Nutzer (Mitarbeiter, Berater, Börsenmakler) können keine Informationen konkurrierender Firmen nutzen;
- in Abschnitt 6 das Gleichgewichtsmodell (Grimm, 1994) mit dem Sicherheitsziel des fairen Austauschs von Ware gegen Geld in offenen Netzen: entweder jeder oder kein Kooperationspartner erreicht sein Kooperationsziel.

Es gibt zahlreiche weitere IT-Sicherheitsmodelle für die verschiedensten Sicherheitsziele, diese vier sind daher nur als typische Beispiele zu verstehen.

Das berühmteste und älteste dieser vier Modelle ist das Bell-LaPadula-Modell, das streng formalisiert vorliegt und das bis heute von den Kriterienkatalogen als eine mögliche Sicherheitsanforderung höchster Vertrauenswürdigkeit geführt wird, obgleich die Zeit der Großrechner, die von vielen Teilnehmern gemeinsam genutzt werden, längst vorbei ist. Das

Clark-Wilson-Modell war eine Art Emanzipation vom Bell-LaPadula-Modell und stellt die Integritätsanforderung an geschäftlich genutzte IT-Systeme heraus. Sein Verdienst liegt in der Einführung der handelnden Subjekte in die Sicherheitsmodellierung und verdeutlicht damit die Grenzen der Automatisierung der Sicherheit. Das Chinese-Wall-Modell, das sicherlich den schönsten Namen trägt, ist wegen seiner sehr speziellen Anwendung praktisch unbedeutend, hat aber für die Theorie der Modellbildung insofern einen hohen Stellenwert, als es das erste Modell war, in dem sich die Wahlfreiheit der Teilnehmer mit zunehmender Nutzung des Systems dynamisch ändert. Das Gleichgewichtsmodell schließlich betrifft zwar mit dem E-Commerce einen sehr bedeutenden Bereich der Internetnutzung, und de facto orientieren sich die Systeme auch daran, aber als explizites Prüfkriterium ist es nie zum Einsatz gekommen. Es zeigt beispielhaft, wie gewisse Sicherheitsziele (fairer Tausch) erreicht werden können, ohne dass man das Wohlverhalten der Kooperationspartner im E-Commerce vorauszusetzen braucht, was ja eine realistische Einschätzung der Sicherheitslage im Internet darstellt: man weiß nie, mit wem man es auf der anderen Seite des Netzes zu tun hat.

	Übergeordnetes Sicherheitsziel	*Sichere Systemzustände*	*Regeln für erlaubte Systemübergänge*
Bell-LaPadula	Vertraulichkeit: vertrauliche Informationen gelangen nicht in die Hand unautorisierter Personen	Personen und Objekte befinden sich in Vertraulichkeitsklassen, die zueinander passen	Zugriffskontrollregeln aufgrund Klassenzugehörigkeit (*no-read-up*, *no-write-down*)
Clark-Wilson	Integrität: das Anwendungssystem repräsentiert seine Anwendung korrekt	Daten und Prozesse sind integer; sensible Prozesse unterliegen der Aufgabenteilung	Nutzung von Verifikations- und Transaktionsprozeduren und ihre Überprüfung
Chinese Wall	Verhinderung von Insiderhandel: Nutzer können keine Informationen konkurrierender Firmen lesen	Subjekte haben keinen Lesezugriff auf Objekte verschiedener konkurrierender Firmen	Je eine Lesezugriffs- und eine Schreibzugriffsregel
Gleichgewicht	Erfolgskopplung: entweder jeder oder kein Partner erreicht sein Kooperationsziel	Gleichgewicht zwischen Verpflichtungen und Beweisen	Sammeln und Abgeben von Beweisen bei Erfüllung bzw. Übernahme von Verpflichtungen

Tabelle 1: Überblick über die vier Sicherheitsmodelle „Bell-LaPadula“, „Clark-Wilson“, „Chinese Wall“ und „Gleichgewicht“.

3. Bell-LaPadula (1973)

Idee:

Das erste in der Literatur bekanntgewordene IT-Sicherheitsmodell stammt von den beiden Mathematikern D. Elliott Bell, und Leonard J. LaPadula von der MITRE Corporation (Massachusetts, USA) 1973. Es bezieht sich auf die damals in Behörden und Firmen übliche Situation von Großrechnern, die die Daten vieler Nutzer halten und verarbeiten. Es war bis dahin üblich, die Zugriffsrechte der Nutzer einzeln festzulegen in einer so genannten Lampson-Matrix. Die Zugriffsmatrix vermerkt für jeden Nutzer und für jedes Objekt einzeln, auf welche Daten welcher Nutzer Lese-, Schreib- und Ausführungsrecht hat. Diese Art des Zugriffschutzes wird wegen der Unterscheidung der einzelnen („diskreten“) Subjekte und Objekte „Discretionary Access Control (DAC)“ genannt. Die klassische Zugriffskontrolle in Unix, zum Beispiel, ordnet zwar jedem Nutzer die Objekte zu, auf die er mit *read*, *write* oder *execute* zugreifen darf, aber seine Objekte kann der Besitzer nicht frei Nutzern mit Zugriffsrechten zuordnen, sondern er kann den Zugriff auf seine Objekte nur für sich selbst,

für seine Gruppe und für „alle anderen“ einstellen. Allerdings lässt sich im Linux-Kern ein Zugriffsverfahren mit *Access Control Lists* (ACL) installieren, das das DAC-Zugriffsmodell erfüllt. Mit „setfacl“ kann der Besitzer einer Subdirectory jedes Objekt einzeln beliebigen Nutzern und Gruppen mit ihren Zugriffsrechten zuordnen, und mit „getfacl“ kann man sich diese Zuordnung anzeigen lassen, siehe Abbildung 1.

```
164 > getfacl asg
# file: asg
# owner: tulkas
# group: employee
user::rwx
user:chnee:rwx
group::--x
mask::rwx
other::--x
165 >
```

Abb. 1: Linux-ACL-Kommando „getfacl“ mit Darstellung der diskreten Zugriffsberechtigungen, angewendet auf das File „asg“ des Owners „tulkas“: Owner und User „chnee“ dürfen alles, Gruppe „employee“ des Owners und alle anderen dürfen „execute“. Es könnten hier weitere Zeilen für andere „user:…“ und „group:…“ eingetragen werden.

Die Idee von Bell und LaPadula besteht nun darin, dieses „Einzelzugriffsrecht“ um eine generelle Zugriffsregel, eine so genannte „Mandatory Access Control (MAC)“, zu erweitern, die bei jedem Zugriff eines Subjektes auf ein Objekt durchgesetzt wird. Dazu werden Subjekte und Objekte in Sicherheitsklassen eingeteilt, die in einer Halbordnung zueinander stehen, das heißt, dass je zwei Klassen entweder nicht miteinander vergleichbar sind oder gleich sind oder die eine „höher“ als die andere ist.

Zur Organisation einer Sicherheitsklasse gibt es zwei Sicherheitsmaße. Zum einen gibt es die strikt geordneten Sicherheitsstufen „unbeschränkt<beschränkt<vertraulich<geheim<streng geheim“, zum anderen gibt es Organisationsgruppen. Jeder Teilnehmer und jedes Objekt ist einer Sicherheitsstufe und einer Menge von Organisationsgruppen zugeordnet. Axel darf zum Beispiel „geheime“ Dokumente lesen, aber keine „streng geheimen“, und er gehört sowohl zum Rechenzentrum, als auch zu einer bestimmten Entwicklergruppe.

Eine Sicherheitsklasse ist höher als die andere, wenn ihre Sicherheitsstufe höher ist und ihre Menge der Organisationsgruppen alle Organisationsgruppen der anderen Klasse umfasst. Axel im Beispiel oben wäre demnach in einer höheren Sicherheitsklasse als ein Drucker, der „vertrauliche“ Dokumente drucken darf, aber keine „geheimen“ oder „streng geheimen“, und der zum Rechenzentrum gehört. In Bezug auf diese Sicherheitsklassen wird die Regel festgelegt, dass ein Subjekt nur Objekte niedrigerer oder gleicher Sicherheitsklassen lesen, und nur Objekte höherer oder gleicher Stufe schreiben darf. Axel dürfte zum Beispiel die Ausgaben des eben erwähnten Druckers lesen, aber nicht darauf drucken. Damit wird verhindert, dass irgendwelche Objekte Subjekten niedrigerer oder fremder Sicherheitsstufen zur Kenntnis kommen können. Das soll die Vertraulichkeit in hierarchisch organisierten Arbeitsumgebungen, wie zum Beispiel in Behörden, schützen.

Bewertung:

Wegen der hierarchischen Organisationsform sagt man, dass diese allgemeine Zugriffskontrolle (Mandatory Access Control – MAC) geeignet für militärische Anwendungen (military context) sei, im Gegensatz zum im nächsten Abschnitt besprochenen Clark-Wilson-Modell, dem man die Eignung für geschäftlich orientierte Arbeitsumgebungen (business context) zuspricht. Beim MAC geht es um den Schutz vertraulicher oder gar

geheimer Informationen in hierarchischen Organisationen, die Informationsflüsse „von unten nach oben“, nicht aber „von oben nach unten“ zulassen. Befehlskommunikation lässt sich also mit Bell-LaPadula nicht modellieren.

Das Bell-LaPadula-Modell wird mit Hilfe der Zugriffskontrollmechanismen DAC und MAC durchgesetzt. Daher ist es in seiner Ausführung ein *Zugriffskontrollmodell*. In seiner Wirkung verhindert es aufgrund der MAC den Informationsfluss „von oben nach unten“. Daher nennen es manche Autoren ein *Informationsflussmodell*.

Das Zugriffsmodell, bzw. das Informationsflussmodell von Bell-LaPadula lässt sich in solchen Umgebungen realisieren, in denen der Zugriff auf Daten zentral kontrolliert wird, wie etwa in Großrechnern oder in zentral gesteuerten Rechenzentren. Für einen Einsatz in offenen Netzen, wie dem Internet, müssten alle beteiligten Anwendungen eine gemeinsame Semantik der Sicherheitsklassen, sowie ein gemeinsames Vertrauensmodell mit Zusicherungsdiensten haben. Dazu müsste das Modell selbst erweitert werden, das ist eine Forschungsaufgabe.

Ein praktisches Problem von Bell-LaPadula-gesteuerten Betriebssystemen besteht darin, dass gelegentlich Objekte von einem höheren Subjekt gelesen, verändert und zurückgeschrieben werden. Ein solches Objekt wird dadurch automatisch auf das Level des Subjekts hochgestuft. Objekte können also durch den laufenden Betrieb automatisch hoch-, aber nicht heruntergestuft werden. Dabei wandern mehr und mehr Objekte nach oben und werden dem Zugriff der darunter angeordneten Subjekte entzogen, bis schließlich das System nicht mehr seinen Anwendungszweck erfüllt. Dann müssen die autorisierten Systemadministratoren das System anhalten und von Hand alle Objekte wieder herabstufen, die „unten“ gebraucht werden. Abgesehen davon, dass das automatische Hochstufen von Objekten ein „günstiges Virenklima“ ergibt (Cohen 1987), ist der Zwang zur regelmäßigen manuellen Abstufung unbequem und zeitraubend.

Wenn Subjekte schreibend auf niedrigere Objekte zugreifen, könnte man statt der Hochstufung dieser Objekte die Subjekte für diesen aktuellen Zugriff temporär herunterstufen. Dabei können aber unerwünschte Informationsflüsse von oben nach unten stattfinden, deren Kontrolle durch die ausführenden Personen, die temporär ihre Sicherheitsklasse verlassen, möglicherweise unsicher ist.

Ausführung:

Das **Sicherheitsziel** des Bell-LaPadula-Modells ist die Verhinderung von vertraulichen Informationsflüssen in unzuständige Bereiche. Das wird durch eine Zugriffskontrolle durchgesetzt, die vier im Modell spezifizierte Regeln beachtet (s.u.). In diesem Sinne ist das Bell-LaPadula-Modell in seiner Wirkung ein *Informationsflussmodell* und in seiner Ausführung ein *Zugriffskontrollmodell*.

Subjekte und Objekte erhalten Namen wie S_1, S_2, S_3, … bzw. O_1, O_2, O_3, … Es gibt die Zugriffsarten “ro – read only”, “ap – append”, “rw – read or write”, “ex – execute”, “ca – change attribute”. Eine sogenannte Lampson-Matrix führt in ihren Zeilen alle Subjekte und in ihren Spalten alle Objekte, die im System verfügbar sind. Die Matrixschnittpunkte benennen die erlaubten Zugriffsarten des Zeilensubjekts auf das Spaltenobjekt. Das Matrixelement (3,4) enthält zum Beispiel den Eintrag “ro, ex” um zum Ausdruck zu bringen, dass das Subjekt S_3 auf das Objekt O_4 lesend und ausführend (und sonst nicht) zugreifen darf.

Weiterhin existieren zwei Sicherheitsmaße. Zum einen gibt es die strikt geordneten Sicherheitsstufen C: „unrestricted<restricted<confidential<strictly confidential<secret<top secret“, denen jedes Objekt als „Classification“ und jedes Subjekt als „Clearance“ zugeordnet ist. Wenn zum Beispiel das Subjekt S_3 die „Clearance“ „strictly confidential“ hat, dann darf S_3 grundsätzlich alle Objekte lesen, die als „strictly confidential“ oder schwächer eingestuft

sind; wenn das Objekt O_4 also die Sicherheitsklasse „confidential“ hat, dann darf S_3 O_4 lesen, wenn es aber die Sicherheitsklasse „secret“ hat, dann darf S_3 es nicht lesen.

Für das andere Sicherheitsmaß K sind alle Subjekte und Objekte in Mengen von Organisationsgruppen (Categories) eingeteilt. S_3 könnte zum Beispiel zur Abteilung des Rechenzentrums, sowie zur Entwicklungsgruppe eines elektronischen Ausweises in der ganzen Organisation gehören. Der Drucker O_4 könnte dem Rechenzentrum zugeordnet sein. In diesem Falle wäre S_3 in einer höheren Kategorie als der Drucker O_4, da es allen Gruppen des Objektes (Rechenzentrum) ebenfalls angehört. Wäre der Drucker O_4 aber im Rechenzentrum nur dem Leitungssekretariat zugeordnet, dann wären die Kategorien von S_3 und O_4 nicht vergleichbar, da sie zwar in einer Gruppe (Rechenzentrum) übereinstimmen, in allen anderen aber schließen sie sich gegenseitig aus: S_3 gehört zur Entwicklungsgruppe, O_4 aber nicht, umgekehrt gehört O_4 zum Leitungssekretariat, S_3 aber nicht.

Mit Hilfe der Sicherheitsstufen C und der Kategorien K wird nun jedes Subjekt S und jedes Objekt O „markiert“ (englisch: „Label Function“): f(S) bzw. f(O)=(c,k), wobei c eine Sicherheitsstufe und k eine Menge von Gruppen $\{G_1, G_2, G_3,\ldots\}$ ist, denen das Subjekt, bzw. das Objekt angehört. Zwei Labels f_1 und f_2 sind nun genau dann miteinander vergleichbar, wenn entweder $c_1 \leq c_2$ *und* $k_1 \subseteq k_2$ oder wenn $c_2 \leq c_1$ *und* $k_2 \subseteq k_1$ ist.

Definition: Es seien S und T jeweils ein Subjekt oder Objekt. Es seien $f(S)=(c_1,k_1)$ und $f(T)=(c_2,k_2)$. Dann gilt die Relation $f(S) \leq f(T)$ genau dann, wenn $c_1 \leq c_2$ *und* $k_1 \subseteq k_2$.

Beispiel 1: $f(O_4)$=(“confidential“, {Rechenzentrum}),
$f(S_3)$=(“strictly confidential“, {Rechenzentrum, Entwicklungsgruppe}).

Dann gelten “confidential“ ≤ „strictly confidential“
und {Rechenzentrum} ⊆ {Rechenzentrum, Entwicklungsgruppe},

also gilt $f(O_4) \leq f(S_3)$.

Beispiel 2: $f(O_4)$=(“secret“, {Rechenzentrum}),
$f(S_3)$=(“strictly confidential“, {Rechenzentrum, Entwicklungsgruppe}).

Dann ist zwar “secret“ ≥ „strictly confidential“,
aber nicht {Rechenzentrum} ⊇ {Rechenzentrum, Entwicklungsgruppe},

also sind $f(O_4)$ und $f(S_3)$ nicht vergleichbar.

Beispiel 3: $f(O_4)$=(“confidential“, {Rechenzentrum, Leitungssekretariat}),
$f(S_3)$=(“strictly confidential“, {Rechenzentrum, Entwicklungsgruppe}).

Dann ist zwar “confidential“ ≤ „strictly confidential“,
aber nicht {Rechenzentrum, Leitungssekretariat } ⊆
{Rechenzentrum, Entwicklungsgruppe},

also sind $f(O_4)$ und $f(S_3)$ nicht vergleichbar.

Beispiel 4: $f(O_4)$=(“secret“, {Rechenzentrum, Leitungssekretariat}),
$f(S_3)$=(“strictly confidential“, {Rechenzentrum}).

Dann sind “secret“ ≥ „strictly confidential“
und {Rechenzentrum, Leitungssekretariat} ⊇ {Rechenzentrum},

also gilt $f(O_4) \geq f(S_3)$.

Definition eines Systemzustands: Ein (aktueller) Systemzustand besteht nun aus der Menge aller (aktueller) Zugriffe aller Subjekte auf alle Objekte in einem System, das eine Zugriffskontrollmatrix M und eine Labelfunktion f hat. Präzise: Ein Systemzustand ist ein Vektor (b, M, f) aus den drei Komponenten

- b := Liste aller gegenwärtigen Zugriffe $\{(S_1, O_1, x_1), (S_1, O_2, x_2),\ldots\}$
- M := Zugriffskontrollmatrix (Lampson-Matrix)
- f := Labelfunktion, welche jedem Subjekt und jedem Objekt eine Vertraulichkeitsstufe c und eine Kategorie k von Gruppen zuordnet.

Ein Zustand wird dadurch in einen anderen Zustand überführt, dass eine der Komponenten b, M oder f sich ändert. M oder f ändern sich selten, und auch nur aufgrund von Eingriffen der autorisierten Systemadministration *Tranquility Principle*. Interessant sind die häufigen Änderungen von b „im laufenden Betrieb“. Diese kommen dadurch zustande, dass ein Subjekt S_i einen neuen Zugriffswunsch x'_i auf ein Objekt O'_i hat oder den Wunsch einen bestehenden Zugriff x_i auf ein Objekt O_i aufzugeben. Den Wunsch, einen Zugriff aufzugeben, kann das Betriebssystem jederzeit erfüllen. Die neue Zugriffsliste b' lautet dann $b'=b-\{(S_i,O_i,x_i)\}$. Einen neuen Zugriff (S_i,O'_i,x'_i) muss das Betriebssystem prüfen, und dann verwirft oder gewährt es ihn. Falls es ihn gewährt, lautet die neue Zugriffsliste $b'=b\cup\{(S_i,O_i,x_i)\}$. Der Nachfolgezustand wird durch das Tripel (b',M,f) markiert.

Nun muss das Betriebssystem die **Regeln für erlaubte Zustandsübergänge** kennen, um Zugriffswünsche so zu entscheiden, dass der nachfolgende Systemzustand sicher ist. Die Regeln entnimmt ein Betriebssystem, das nach dem Modell von Bell-LaPadula arbeitet, den vier Bedingungen des wie folgt spezifizierten „sicheren Systemzustands“.

Spezifikation eines sicheren Systemzustands: Ein Systemvektor (b,M,f) ist sicher, wenn für alle Zugriffe $(S,O,x)\in b$ gilt

1. Discretionary Access / Need-to-Know: $x\in M(S,O)$, d.h. die Zugriffsmatrix erlaubt S den x-Zugriff auf O
2. Simple Security / no read-up: (x=ro oder x=rw) $\Rightarrow$ $f(S) \geq f(O)$
3. *-Security / no write-down: (x=ap oder x=rw) $\Rightarrow$ $f(S) \leq f(O)$
4. Tranquility Principle: Nur ein kleiner Kreis von vertrauenswürdigen Systemadministratoren hat Schreibrecht zur Veränderung der Lampson-Matrix M und der Labelfunktion f.

In Beispiel 1 oben wäre der einfache Lesezugriff von S_3 auf O_4 „sicher“, wenn in der Lampson-Matrix M(3,4) ein „ro“ steht, da die Regel 2 der Simple-Security greift, mit anderen Worten: S_3 dürfte die Ausdrücke des Druckers O_4 lesen, aber nicht darauf drucken. In Beispiel 4 oben wäre ein blinder Schreibzugriff von S_3 auf O_4 „sicher“, wenn in der Lampson-Matrix M(3,4) ein „ap“ steht, da die Regel 3 der *-Security greift, mit anderen Worten, S_3 dürfte auf dem Drucker O_4 drucken, die Ausdrucke von O_4 aber nicht lesen. Lesen und schreiben dürfte ein Subjekt S von und auf einem Objekt O nur, wenn alle drei Regeln greifen, wenn also „rw“$\in$M(S,O) und wenn f(S)=f(O).

Mit diesen Begriffsbildungen formuliert das Bell-LaPadula-Modell dann das entscheidende

Sicherheitstheorem: Wenn ein Systemvektor v=(b,M,f) sicher im Sinne der Definition eines sicheren Systemzustands ist und wenn das System erstens nur solche Zugriffe freigibt, die den Regeln 1-3 (*need-to-know*, *no-read-up*, *no-write-down*) entsprechen und wenn sich das System zweitens zum Schutz der Zugriffsmatrix M und der Labelfunktion f an das *Tranquility Principle* hält, dann wird der nachfolgende Systemzustand v'=(b',M',f') ebenfalls sicher im Sinne der Definition eines sicheren Systemzustands sein.

Die Begriffsbildung für einen sicheren Systemzustands ist bereits derart gefasst, dass das Theorem sich wie selbstverständlich daraus ergibt. Dennoch muss man für eine mathematisch saubere Beweisführung einigen Aufwand betreiben. Dazu analysieren die Autoren des Bell-LaPadula-Modells in ihren Artikel (1973 und 1975) die zehn möglichen Zugriffsarten „get–read“, „get–append“, „get–execute“, „get–write“, „release–read/write/all/execute“, „give–read/write/all/execute (change Matrix M)“, „rescind–read/write/all/execute (change Matrix M)“, „change–label“, „create–object“ und „delete–object“. Darauf wird an dieser Stelle verzichtet. Gute Zusammenfassungen des Bell-LaPadula-Modells finden sich in allen Lehrbüchern zur IT-Sicherheit, zum Beispiel bei (Eckert 2006).

Das **Vertrauensmodell** wird von den Autoren Bell und La Padula nicht explizit ausgeführt. Sie setzen aber voraus, dass das Modell nur dann das übergeordnete Sicherheitsziel der Vertraulichkeit erfüllt, wenn die verantwortlichen Systemadministratoren die Zugriffsrechte und Labelfunktionen der Organisationsstruktur korrekt anpassen. Außerdem müssen sich alle beteiligten Personen ihrer organisatorischen Aufgabe gemäß angemessen verhalten, zum Beispiel dürfen sie nicht in der einen Rolle, in der sie sich für einen Schreibvorgang temporär herabstufen lassen, ausplaudern, was sie in einer anderen Rolle vertraulich erfahren haben.

4. Clark-Wilson (1987)

Idee:

David D. Clark vom MIT und David R. Wilson von der IT-Beratungsfirma Ernst & Whinney fanden 1987, dass vertrauliche Informationsflüsse nicht das wichtigste Sicherheitsproblem geschäftlicher IT-Anwendungen bilden. Informationstechnik unterstützt in der Geschäftswelt vor allem geschäftliche Transaktionen durch eine Darstellung von Geld- und Warenflüssen und durch die Erstellung von Aufträgen, Rechnungen und Quittungen. Die Bits und Bytes im Computer müssen die Gegenstände der Welt richtig zum Ausdruck bringen. Wenn in der Datenbank unter „Lagerbestand“ „1,5 Tonnen Mehl“ vermerkt ist, dann muss das Lager auch genau 1,5 Tonnen Mehl enthalten. Allen Bewegungen der geschäftlichen Welt müssen die Zustandsänderungen im Computersystem genau entsprechen. Das Sicherheitsziel geschäftlicher Anwendungen ist daher der Schutz der Integrität der Anwendungsprozesse und ihrer Daten.

Nun kann man zwar durch Umdrehen der „Größer-Kleiner-Relationen“ im Bell-LaPadula-Modell die Integrität der Daten dadurch schützen, dass nur „höhere Subjekte“ – diese entsprechen vertrauenswürdigen Personen und Prozessen – „niedrigere Objekte“ verändern dürfen. Aber damit ist nach Auffassung von Clark und Wilson die Beziehung zwischen Wirklichkeit und Informationssystem nicht ausreichend klar beschrieben. Ihr Verdienst ist es, die handelnden Subjekte mit ihrer Anfälligkeit zu Fehlern und Betrug explizit im Modell zum Ausdruck zu bringen. Über so genannte „wohlgeformte Transaktionen“ der Computerdaten hinaus führen sie zur Behebung von Fehlern im Betriebsablauf Verifikationsprozeduren ein. Das sind „Inventuraktivitäten“, in denen die Daten und Prozesse der Computer noch einmal explizit gegenüber der Realität abgeprüft und ggf. verbessert werden. Zur Vermeidung von Fehlern und von Betrug führen sie zusätzlich noch das Prinzip der Aufgabenteilung (Separation of Duty) ein, nach dem alle sensiblen Transaktionen von zwei voneinander unabhängigen Personen gemeinsam ausgeführt werden müssen, weil das „erfahrungsgemäß“ das Risiko des Fehlverhaltens Einzelner vermindert.

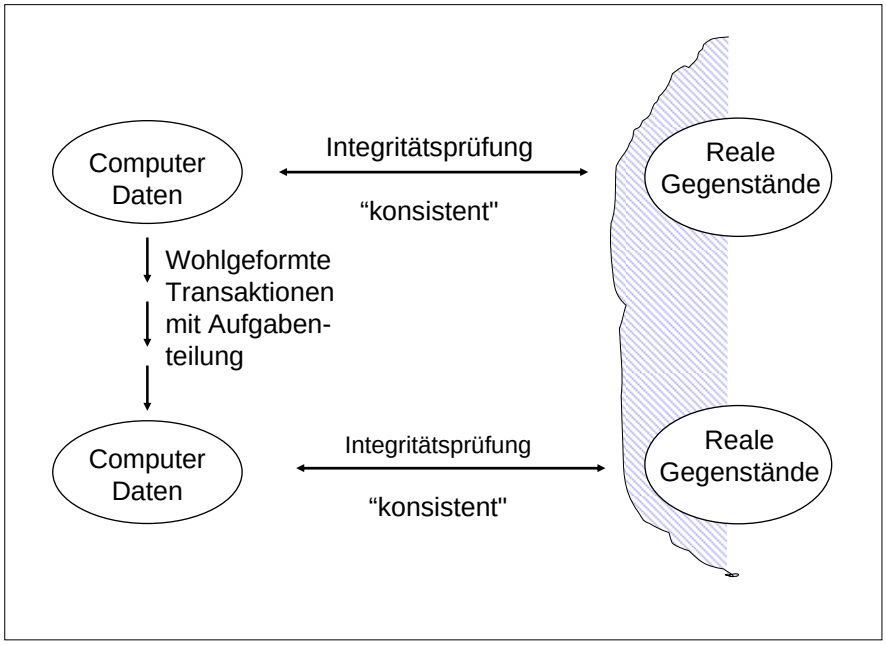

Abbildung 2: Modellskizze des Clark-Wilson-Integritätsmodells

Der Kern des Modells ist die Spezifikation einer Menge von Zertifizierungs- und Durchsetzungsregeln. Die Zertifizierungsregeln werden von den Sicherheitsverantwortlichen ausgeführt, die Durchsetzungsregeln vom System. Die Zertifizierungsregeln, gemäß denen die Sicherheitsverantwortlichen eine korrekte Auslegung des Systems überprüfen, behandeln wohlgeordnete Transaktionen, Verifikationsprozeduren und das Aufgabenteilungsprinzip. In den Durchsetzungsregeln sorgt das System (automatisch) für die Einhaltung der zertifizierten Eigenschaften, darunter für die Zuordnung von Prozeduren zu verantwortlichen Personen, die nach der Aufgabenteilung in Unverträglichkeitsklassen eingeteilt sind. Zum Beispiel darf kein Systemadministrator diejenigen Prozeduren, deren Rechte er verwaltet, selber ausführen. Solche Regeln müssen spezifiziert sein, dann kann das System sie auch überwachen und durchsetzen.

Ein sicherer Systemzustand ist nun ein Zustand, in dem erstens alle Modellregeln implementiert sind, in dem aber darüber hinaus die Verifikationen und die Aufgabenteilung durch die Systemverantwortlichen sorgfältig (und das heißt nicht automatisch, sondern nach menschlichem Ermessen!) ausgeführt werden. In diesem Fall, so die These des Modells, stellt ein IT-System das von ihm unterstützte Geschäft korrekt dar.

Bewertung:

Das Clark-Wilson-Modell beschreibt Mechanismen, die ein Computersystem darin unterstützen, seine geschäftliche Anwendung korrekt darzustellen. Es ist in seiner ursprünglichen Form informell. Es wurde nachträglich von anderen Autoren formalisiert (Terry/Wiseman 1989) und in dieser Form auch für die Sicherheitszertifizierung von Mikroprozessoren verwendet, hat aber für die Sicherheitszertifizierung nie die gleiche Bedeutung erlangt wie das Bell-LaPadula-Modell. Sein Verdienst liegt vielmehr in der expliziten Einbeziehung der handelnden Menschen in das Sicherheitsmodell. Verifikationsprozeduren und Aufgabenteilung setzen der Automatisierung von Sicherheit insofern eine Grenze, als ihre Sicherheit im menschlichen Verhalten liegt: Verifikationsprozeduren erlauben Menschen, Computerdaten mit der von ihnen dargestellten Wirklichkeit zu vergleichen, damit können Betriebsfehler behoben werden. Und die

Aufgabenteilung nutzt die Erfahrung, dass zwei Menschen seltener denselben Fehler begehen bzw. sich zu einem Betrug zusammenfinden. Es heißt in ihrem Artikel wörtlich: “Verification and certification, although related to the internal consistency, are not strictly internal procedures, but can only be performed with respect to a specific integrity policy related to the real world.” (Clark/Wilson 1987)

Der Kern des Modells ist die Spezifikation einer Menge von Zertifizierungs- und Durchsetzungsregeln, die von Sicherheitsverantwortlichen, bzw. vom System zentral für die gesamte Anwendung ausgeführt werden. In diesem Sinne ist es ein *Transaktionsmodell*. Damit beschreibt es die Sicherheit einer Geschäftsanwendung aus Sicht eines Marktteilnehmers. Das Zusammenspiel autonomer Partner kann es nicht zum Ausdruck bringen, das geschieht erst durch offene Regeln, wie im Gleichgewichtsmodell (s.u.).

Ausführung:

Das **Sicherheitsziel** des Clark-Wilson-Modells ist die korrekte Darstellung der Geschäftsanwendung durch das System. Es wird durch eine Menge von wohldefinierten Transaktionen durchgesetzt, die auf eine geprüfte Situation des Systems aufsetzen. In diesem Sinne ist das Clark-Wilson-Modell ein *Transaktionsmodell*.

Das Clark-Wilson-Modell verlangt von einem IT-System, dass es Nutzer, Daten und Aktionen in bestimmter Weise explizit führt: Es gibt eingeschränkte Daten (Constrained Data Items – CDI) und (noch) uneingeschränkte Daten (Unconstrained Data Items – UDI). Eingeschränkte Daten sind zum Beispiel Datenbankeinträge, die einen Lagerbestand beschreiben. Uneingeschränkt könnten Daten sein, die in einem Lieferbericht über neu eingetroffene Waren von außen enthalten sind. Indem die gelieferten Waren in das Lager übernommen werden, müssen die Lieferdaten geprüft und in die Datenbank des Lagerbestandes übernommen werden und werden dadurch zu eingeschränkten Daten.

Es gibt weiterhin wohlgeformte Transaktionen, denen die Menge der erlaubten Inputdaten und die Menge der erlaubten ausführenden Nutzer explizit zugeordnet sind. Die berechtigten Nutzer der wohlgeformten Transaktionen unterliegen der Aufgabenteilungsregel. Schließlich ist jede einzelne Systemtransaktion zu protokollieren, einschließlich der ausführenden Nutzer und der Input- und Outputdaten.

Auf dieser Basis formuliert das Modell fünf Zertifizierungsregeln und vier Durchsetzungsregeln. Die Zertifizierungsregeln C1-C5 werden von Sicherheitsverantwortlichen aufgrund einer vom Anwender zu spezifizierenden Sicherheitsrichtlinie (Security Policy) ausgeführt: sie überprüfen und zertifizieren den formulierten Sachverhalt. Die Durchsetzungsregeln E1-E4 werden vom System automatisch ausgeführt. Die Regeln sind nach Themen geordnet.

Interne Konsistenz:

C1: Es werden Prozeduren zur Verifikation der Integrität (Integrity Verification Procedures – IVP) eingeführt. Diese stellen fest, dass alle eingeschränkten Daten (CDI) gültig sind, d.h. dass sie ihre Beschreibungsgegenstände korrekt wiedergeben.

C2: Alle Transaktionsprozeduren (TP) sind gültig. Das bedeutet, dass sie alle gültigen CDI in einen gültigen Zustand überführen (TP1,(CDI_a,CDI_b,CDI_c,….)).

E1: Das System führt eine Zuordnungsliste aus Regel C2 und sorgt dafür, dass eine TP nur gemäß dieser Liste auf ein CDI zugreifen darf.

Externe Konsistenz:

E2: Das System führt eine Zuordnungsliste von Nutzern auf TPs (User X, TP_i, (CDI_a,CDI_b,CDI_c,…)) und sorgt dafür, dass Nutzer nur gemäß dieser Liste TPs ausführen dürfen.

C3: Die Zuordnungsliste aus Regel E2 folgt dem Aufgabenteilungsprinzip.

E3: Das System authentifiziert vor Ausführung einer TP die Identität des Nutzers.

Mitschnitt/Protokollierung:

C4: Alle TP schreiben in eine Logdatei alle Informationen, die erforderlich sind, das Vorgehen der TP zu rekonstruieren. Die Logdatei ist selbst ein eingeschränktes Datenobjekt (CDI), in das nur Daten hinzugefügt, aber niemals korrigiert oder gelöscht werden können.

Einführung von eingeschränkten Daten:

C5: Jede TP, das ein uneingeschränktes Datum (UDI) in ein eingeschränktes Datum (CDI) überführt, ist so geartet, dass es entweder ein UDI in ein gültiges CDI überführt oder keine Transformation ausführt. Solche TPs sind typischerweise Edit-Programme.

Trennung von Rechtevergabe und Programmausführung:

E4: Nur diejenigen Subjekte dürfen die Zuordnungslisten zwischen Nutzern, TPs und CDIs ändern, die das Recht haben, diese nach Regeln E1 und E2 zu zertifizieren. Diese Subjekte dürfen aber nach der Aufgabenteilungsregel die zugehörigen TPs nicht selbst ausführen.

Spezifikation eines sicheren Systemzustands: Ein System ist in einem sicheren Zustand, wenn es nach den Regeln C1-C5 positiv zertifiziert ist und wenn (irgendwann zuvor) die Verifikationsprozeduren aus der Regel C1 mit positivem Ergebnis ausgeführt worden sind.

Ein **erlaubter Zustandsübergang** wird durch jede Transaktion, die nach den Regeln E1-E4 stattfindet, erzeugt.

Das **Sicherheitstheorem** von Clark und Wilson lautet nun, dass ein System, das nach den Regeln C1-C5 positiv zertifiziert ist und nur Transaktionen nach den Regeln E1-E4 zulässt, jederzeit bei einer Prüfung der Regeln C1-C5 wieder positiv zertifiziert wird. Diese Aussage ist teilweise formalisierbar.

Das **Vertrauensmodell**, das von den Autoren Clark und Wilson ausgiebig diskutiert wird, verlangt nun, dass die Systemdesigner die Transaktionsprozeduren, die Verifikationsprozeduren und die Gültigkeit der Daten richtig spezifizieren und dass die Organisationsverantwortlichen das Aufgabeteilungsprinzip realistisch umsetzen. Unter diesen Voraussetzungen, so lautet die These des Vertrauensmodells, gilt dann: Wenn ein System nach den Regeln C1-C5 positiv zertifiziert ist und gelegentlich die Verifikationsprozeduren aus der Regel C1 ausführt, dann stellt es seine Geschäftsanwendung korrekt dar. Diese Aussage ist nicht formalisierbar.

Genau genommen lässt das Modell sogar zu, dass es trotz Beachtung aller Regeln zu unvorhergesehen Inkonsistenzen kommt. Es sieht nämlich in unregelmäßigen Abständen die Ausführung von Prozeduren zur Verifikation und Wiederherstellung der Integrität nach Regel C1 vor. Damit konzediert das Modell insbesondere ein gelegentliches Versagen des Aufgabenteilungsprinzips: Auch mehrere Nutzer zusammen könnten einen Betrug gemeinsam ausführen oder trotz erhöhter Aufmerksamkeit einen Fehler machen. Deshalb ist der Erfolg dieses Modells nicht rein mathematisch zu beweisen. Es versteht sich daher selbst bei formaler Auslegung als semi-formales Modell.

5. Chinese Wall (Nash-Brewer, 1989)

Idee:

Das Modell der Chinesischen Mauer zielt auf einen Konkurrentenschutz im Börsenumfeld. Genauer: In der Londoner City sollte eine von allen Firmen gemeinsam getragene Zugriffskontrolle dafür sorgen, dass Unternehmensberater, die für verschiedene Firmen derselben Branche tätig sind, kein Insiderwissen in konkurrierende Firmen tragen oder gar zu betrügerischem Börsenhandel ausnutzen können. Das Modell wurde 1989 von Brewer und Nash auf der renommierten IEEE-Konferenz „Security and Privacy“ vorgestellt. Es verdankt seinen schönen Namen der Vorstellung des Mauerbaus: Am Anfang gibt es keine Mauer, jeder Nutzer ist in seinen Entscheidungen frei. Sobald aber ein Nutzer sich für eine Tätigkeit bei einer Firma entschieden hat, verbaut er sich die Möglichkeit, für andere Firmen derselben Branche tätig zu werden.

Ein sicherer Systemzustand ist dadurch gekennzeichnet, dass ein Nutzer keinen Zugriff auf Daten mehrerer konkurrierender Firmen hat. Er wird durch die Ausführung von Lese- und Schreibregeln überprüft und durchgesetzt. Es gibt je eine Lese- und eine Schreibregel.

Die Leseregel lautet so: Ein Subjekt darf ein Objekt nur dann lesen, wenn es bisher nur solche Objekte gelesen hat, die entweder zu derselben Firma gehören oder zu einer anderen Firma einer anderen Branche gehören oder die öffentlich zugänglich sind.

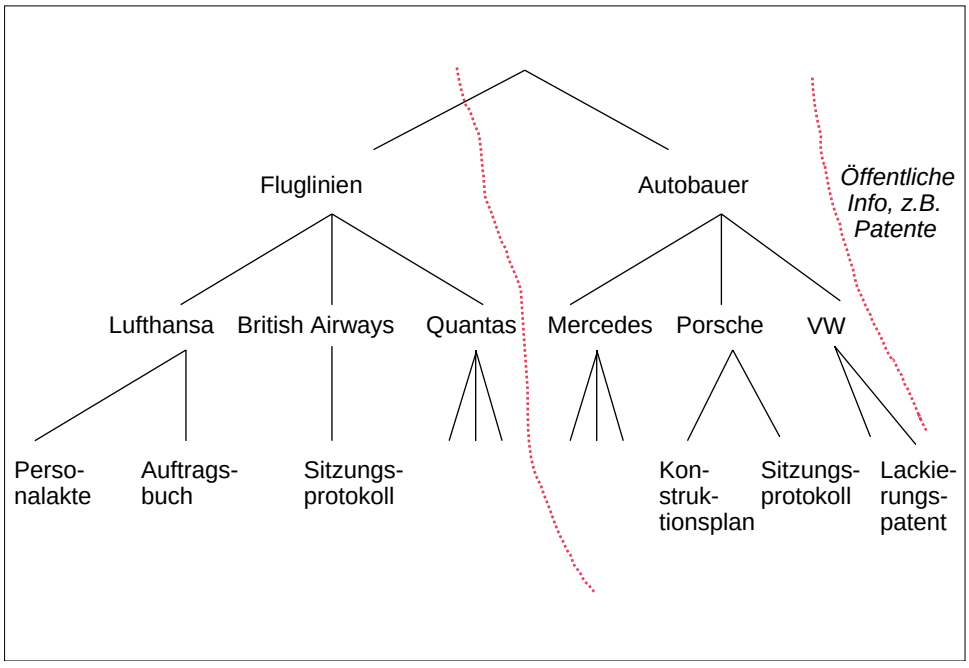

Abbildung 3: Beispiel für einen Chinese-Wall-Objektbaum

Leider reicht die Leseregel zur Durchsetzung des Modellziels nicht aus. Es könnte nämlich sein, dass durch mehrfaches Schreiben über Branchengrenzen hinweg doch ein Informationsfluss zwischen konkurrierenden Firmen zustande kommt. Dazu ein Beispiel: Axel ist ein Berater von Fluglinien und liest Daten der Lufthansa. Axel darf zusätzlich Berater der Autobauerbranche bei Porsche werden. Die Leseregel verhindert nicht, dass Axel die Daten der Lufthansa in Dateibestände von Porsche schreibt. Berta ist ebenfalls eine Beraterin von Porsche und berät zusätzlich die Fluglinie British Airways. Berta könnte nun ohne Verletzung der Leseregel die vom ersten Berater Axel bei Porsche hinterlegten Informationen über die Lufthansa weiter zu British Airways tragen, denn die Leseregel erlaubt ihr, Daten

von Porsche zu lesen und in Datenbestände von British Airways zu schreiben, da diese Firmen untereinander branchenfremd sind. Indirekt ist es auf diese Weise zu einem unerwünschten Informationsfluss von der Lufthansa über Porsche zu British Airways innerhalb derselben Branche der Fluglinien gekommen. Um das zu verhindern, muss zusätzlich eine Schreibregel eingeführt werden.

Die Schreibregel lautet so: Ein Subjekt darf auf einem Objekt nur dann schreiben, wenn es bisher nur solche Objekte gelesen hat, die zur selben Firma gehören oder eine öffentlich zugängliche Information darstellen. Mit anderen Worten, ein Subjekt darf niemals Objekte anderer Firmen gelesen haben, selbst wenn sie zu einer anderen Branche gehören. Das ist eine sehr weitreichende Einschränkung für Personen. Faktisch schließt das eine Beratungstätigkeit für mehr als eine Firma aus, denn unter Beratung versteht man ja wohl nicht nur Lesen, sondern auch Meinungsäußerung, und das ist informatorisch ein Schreibvorgang.

Das Sicherheitstheorem des Modells der Chinesischen Mauer besagt nun trivialerweise, dass bei Einhaltung der Lese- und Schreibregel kein Informationsfluss zwischen verschiedenen Firmen derselben Branche stattfinden kann.

Zur Durchsetzung der Lese- und Schreibregel muss nicht nur eine allgemeine Lampson-Matrix mit diskreten Zugriffsrechten geführt werden, in der nach dem im Bell-LaPadula-Modell formulierten DAC-Verfahren individuelle Lese- und Schreibzugriffe geregelt werden. Sondern zusätzlich wird eine dynamisch wachsende Protokollmatrix geführt, die die stattfindenden Zugriffe laufend aufnimmt und die zur Prüfung der Lese- und Schreibregel konsultiert wird, bevor die Erlaubnis zu einem neuen Zugriff erteilt wird.

Bewertung:

Das Modell erfordert eine zentrale Kontrolle. Alle Firmen aller Branchen, die ihre Berater dem Modell der Chinesischen Mauer unterwerfen, müssen eine solche Zugriffskontrolle gemeinsam ausführen lassen. In diesem Sinne ist das Chinese-Wall-Modell ein *Zugriffskontrollmodell*.

Das Aufregende des Modells der Chinesischen Mauer ist seine Dynamik. Mit zunehmender Dauer wird die Mauer größer, da mit einer aufgenommenen Tätigkeit andere Tätigkeiten verboten werden. Darin liegt das erste praktische Problem des Modells. In Wirklichkeit sollte sich die Mauer an den Stellen wieder abbauen, an denen Tätigkeiten aufgegeben werden und Sperrfristen auslaufen. Das kann man in der Praxis durch manuelle Bereinigung der Protokollmatrix erreichen.

Bei strenger Auslegung der Schreibregel gäbe es keine Dynamik, da ein Berater doch nur eine einzige Firma beraten kann. Wenn man aber das Modell nicht streng auslegt, indem man nur die Leseregel, aber nicht die Schreibregel exekutiert, dann ist zwar ein indirekter Informationsfluss nicht systematisch ausgeschlossen, aber er kann durch Überprüfung der dynamischen Zugriffsprotokollmatrix wenigstens nachträglich festgestellt werden. Insofern erfüllt das dynamische Zugriffsmodell der Chinesischen Mauer, das sich auf die Leseregel beschränkt, doch einen praktischen Zweck.

Ausführung:

Das **Sicherheitsziel** des Chinese-Wall-Modells ist die Verhinderung von Insiderhandel: Nutzer können keine Informationen konkurrierender Firmen nutzen. Es wird durch zwei Zugriffsregeln zum Lesen und Schreiben durchgesetzt und ist deshalb ein *Zugriffskontroll-modell*.

Das Modell der Chinesischen Mauer enthält Subjekte, Objekte und Lese- und Schreibzugriffe. Objekte sind durch die x-Funktion Branchen und durch die y-Funktion Firmen zugeordnet.

Zum Beispiel sei das Objekt o1 die Personalakte von Charlotte in der Firma Lufthansa, dann wäre $x(o_1)$=Fluglinien und $y(o_1)$=Lufthansa. Der Index 0 bezeichnet die öffentliche Zugänglichkeit, Objekte mit den Zuordnungen x_0 oder y_0 sind öffentlich zugänglich und daher „ungefährlich“. Das Lackierungspatent o6 der Firma VW zum Beispiel ist natürlich öffentlich zugänglich und hat daher die Zuordnungen $x(o_6)=x_0$ und $y(o_6)=y_0$.

Spezifikation eines sicheren Systemzustands: Ein Systemzustand ist die Menge aller Zugriffe aller Subjekte auf alle Objekte aller beteiligten Firmen. Ein Systemzustand ist *sicher*, wenn kein Subjekt Lesezugriff auf ein Objekt hat, das zu einer anderen Firma derselben Branche gehört oder zuvor gehört hat. Um sichere Systemzustände zu erreichen, werden statische und dynamische Zugriffskontrollmatrizen geführt und zur Durchsetzung einer Lese- und einer Schreibregel genutzt.

Es gibt zwei Zugriffskontrollmatrizen M und N(t). M ist die bekannte statische Lampson-Matrix, die die diskreten Zugriffsrechte speichert. Der Personalsachbearbeiter Dirk der Firma Lufthansa, der die Personalakte o_1 von Charlotte bearbeiten darf, hat in der Lampson-Matrix an der zugehörigen Stelle M(Dirk, o_1)={read, write} stehen. Für den Berater Axel von Lufthansa aber, der diese Personalakte lesen, aber nicht schreiben darf, steht in der Lampson-Matrix M(Axel, o_1)={read}. Wenn Axel zum Zeitpunkt „27.Januar 2008, 17:23“ o_1 liest, dann steht in der dynamischen Zugriffsprotokollmatrix N(t)(Axel, o_1)=∅, wenn t<„27.Januar 2008, 17:23“, und N(t)(Axel, o_1)={read}, wenn t≥„27.Januar 2008, 17:23“. Nach diesem Datum werden also weitere Lesezugriffe von Axel auf andere Objekte mit $x(o_1)$ und $y(o_1)$ zu vergleichen sein.

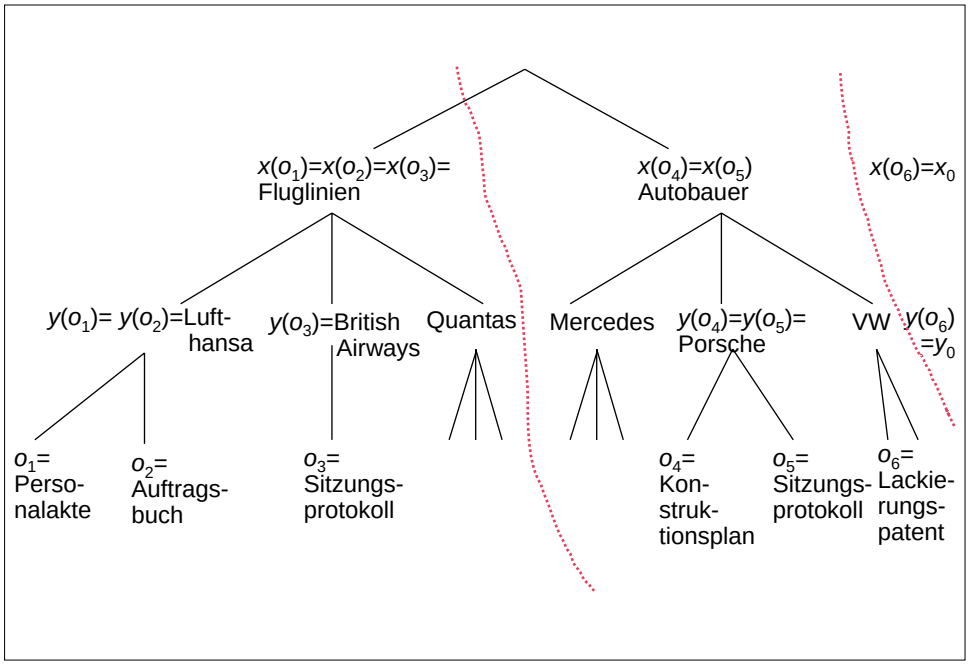

Abbildung 4: Chinese-Wall-Objektbaum mit x- und y-Funktion

Leseregel: Ein Subjekt $s \in S$ darf auf ein Objekt $o \in O$ mit $r \in \{read, execute\}$ zugreifen, genau dann wenn

$$r \in M(s,o) \wedge (\forall o' \in O \text{ gilt: } N(t)(s,o') \neq \varnothing \Rightarrow y(o')=y(o) \vee x(o') \neq x(o) \vee y(o')=y0)$$

Deutung: Ein Subjekt darf ein Objekt nur dann lesen, wenn es erstens die Berechtigung dazu hat (r∈M(s,o)) und wenn es bisher nur auf solche Objekte zugegriffen hat (N(t)(s,o‘) ≠ ∅), die

entweder zu derselben Firma gehören (y(o‘)=y(o)) oder zu einer anderen Firma einer anderen Branche(x(o‘)≠x(o)) gehören oder die öffentlich zugänglich sind (y(o‘)=y0).

Die Schreibregel, die einen indirekten Informationsfluss verhindern soll, lautet

Schreibregel: Ein Subjekt $s \in S$ darf auf ein Objekt $o \in O$ mit *write* zugreifen, genau dann wenn

$write \in M(s,o) \wedge (\forall o' \in O \text{ gilt: } read \in N(t)(s,o') \Rightarrow y(o')=y(o) \vee y(o')=y0)$

Deutung: Ein Subjekt darf in einem Objekt nur dann schreiben, wenn es erstens die Berechtigung dazu hat (write∈M(s,o)) und wenn es bisher nur solche Objekte gelesen hat (read∈N(t)(s,o‘)), die entweder zu derselben Firma gehören (y(o‘)=y(o)) oder die öffentlich zugänglich sind (y(o‘)=y0).

Erlaubte Zustandsübergänge: Ein Zustandsübergang findet durch einen neu hinzukommenden oder wegfallenden Zugriff eines Subjekts auf ein Objekt statt. Ein Zugriffswunsch wird erlaubt, wenn er die Lese- und Schreibregel befolgt.

Sicherheitstheorem: Bei Einhaltung der Lese- und Schreibregel kann keine Information zwischen Firmen derselben Branchen fließen. Bei Einhaltung der Leseregel allein kann Information im Zusammenspiel mindestens zweier Berater durch zweimaliges Überschreiten der Branchengrenzen hinweg fließen, was in der dynamischen Zugriffsprotokollmatrix N(t) nachvollziehbar wäre.

Der Beweis des ersten Teils des Theorems ist trivial. Beispiele für den indirekten Informationsfluss im zweiten Teil sind ebenfalls leicht zu finden, siehe das Beispiel mit Axel und Berta oben. Die zugehörige Protokollspur in der dynamischen Zugriffsprotokollmatrix N(t) sähe dabei so aus: Zum Zeitpunkt t_1 liest Axel das Auftragsbuch o_2 mit $y(o_2)$=Lufthansa, zum Zeitpunkt $t_2>t_1$ schreibt Axel o_2 als o_2’ mit $y(o_2')$=Porsche, zum Zeitpunkt $t_3>t_2$ liest Berta dieses o_2’ und schreibt es zum Zeitpunkt $t_4>t_3$ als o_2’’ mit $y(o_2'')$=British Airways. Achtung, nun gelten $y(o_2)\neq y(o_2'')$ und $x(o_2)=x(o_2'')$, wobei die Objekte dieselbe Information enthalten. Die dynamische Zugriffsprotokollmatrix N(t) enthält nun nach den jeweiligen Zugriffszeitpunkten diese Einträge: $N(t>t_1)$(Axel, o_2)={read}, $N(t>t_2)$(Axel, o_2')={write}, $N(t>t_3)$(Berta, o_2')={read} und $N(t>t_4)$(Berta, o_2'')={write}. Die beiden Einträge in N(t) von Axel widersprechen nicht der Leseregel, wohl aber der Schreibregel! Er könnte es hinterher sogar ohne Verletzung der Leseregel in beiden Firmen wieder lesen, da $x(o_2)\neq x(o_2')$. Ebenso wenig widersprechen die beiden Einträge in N(t) von Berta der Leseregel, wohl aber der Schreibregel. Jeder weitere Lesezugriff auf ein Objekt o von British Airways (y(o)=British Airways) eines Beraters von British Airways ist nunmehr mit dem Leszugriff auf o_2’’ verträglich, da nunmehr $y(o)=y(o_2'')$. Das Problem liegt darin, dass o_2’’ Informationen von o_2 enthält und o_2 zu einer anderen Firma derselben Branche ($y(o_2'')\neq y(o_2)$ und $x(o_2'')=x(o_2)$) gehört. Wer diesen Informationsfluss aufdecken will, muss also den inhaltlichen Zusammenhang zwischen o_2 und o_2’’ aufdecken, dann ist der Weg von o_2 über o_2’ zu o_2’’ nachvollziehbar und das Zusammenspiel zwischen Axel und Berta erkennbar.

Bei einer Beachtung der Schreibregel wäre aber Axels Lesen bei Lufthansa und sein Schreiben bei Porsche ebenso wenig zugelassen worden wie Bertas Lesen bei Porsche und ihr Schreiben bei British Airways. Die Schreibregel ist so streng, dass sie eine praktische Beratungstätigkeit, die Schreiben erfordert, für mehr als eine Firma auch dann verhindert, wenn die Firmen verschiedenen Branchen angehören. Die Schreibregel verbietet ja bereits das schlichte Lesen in Daten anderer Firmen. Daher wird die Schreibregel nicht automatisch exekutiert. Das Chinese-Wall-Modell nutzt in der Praxis nur die Leseregel und nimmt die Möglichkeit des indirekten Informationsflusses in Kauf. Ein unerwünschter Informationsfluss

kann dann wenigstens nachträglich durch Prüfung der dynamischen Zugriffsprotokollmatrix festgestellt werden.

Das **Vertrauensmodell** des Chinese-Wall-Modells formuliert die Anforderungen an das Anwendungssystem, die es erfüllen muss, damit keine Information an der Lese- und Schreibregel vorbei fließen kann. Erstens müssen alle beteiligten Firmen in Bezug auf eine gemeinsam anerkannte und durchgesetzte Zugriffskontrolle miteinander kooperieren. Zweitens müssen alle beteiligten Beratungspersonen ihren User-Ids im System korrekt und eindeutig zugeordnet werden. Das Vertrauensmodell kann außerdem die Sicherheitslücke schließen, die bei der Lockerung der Schreibregel entsteht, indem es die Überprüfung der dynamischen Zugriffsprotokollmatrix regelt.

6. Gleichgewicht (1993)

Idee:

Das Gleichgewichtsmodell wurde erstmals auf der 16th National Computer Security Conference vorgetragen (Grimm 1993) und war das erste, das den fairen Austausch von digitalen Gütern zwischen autonomen Partnern im offenen Internet beschreibt. In der Nachfolge von Clark und Wilson bezieht es die handelnden Akteure in das Modell mit ein, wobei aber hier die Kooperationspartner in einem offenen Netz agieren. Das heißt, dass die Akteure sich zwar auf ihr eigenes Verhalten, nicht aber auf das Verhalten ihrer Partner verlassen können: sie müssen damit rechnen, dass sich die andere Seite nicht wohl verhält. Das Gleichgewichtsmodell verzichtet also in Abkehr von bisherigen Sicherheitsmodellen auf zentral gesteuerte Kontrollmechanismen. Es beantwortet die Frage, wie man in dieser Situation sicher über das Internet Handel treiben oder überhaupt verbindlich kommunizieren kann.

Die Lösung liegt in einer geeigneten Ausgestaltung einer Kooperation, in der die Verpflichtungen wechselseitig „Zug um Zug“ erfüllt werden, und in der richtigen Abarbeitung der Züge, indem für alle eingegangenen und eingelösten Verpflichtungen Beweise geliefert werden: Verpflichtungen und Beweise müssen „im Gleichgewicht“ gehalten werden. Das Gleichgewichtsmodell legt Regeln fest, nach denen Beweise zu liefern und einzufordern sind. Als Zeichen des eigenen Wohlverhaltens sendet ein Akteur seine signierten Willenserklärungen und Empfangsbestätigungen zum gehörigen Zeitpunkt an seinen Partner. Falls von der anderen Seite zum erwarteten Zeitpunkt kein angemessener Beweis geliefert wird, bricht ein Akteur die Kommunikation ab und fordert Kompensation oder Fortsetzung aufgrund der vorhandenen Beweislage. Zu diesem Zweck müssen die Beweise auch außerhalb des Netzes justitiabel sein.

Das *Sicherheitsziel* des Gleichgewichtsmodells ist die „Erfolgskopplung“ der Kooperation: entweder haben beide Partner Erfolg oder keiner von beiden. Zum Beispiel darf es in einer einfachen Verkaufskooperation niemals passieren, dass der Verkäufer das Geld erhält, der Käufer aber nicht die Ware, oder umgekehrt, dass der Käufer die Ware erhält, der Verkäufer aber nicht das Geld.

Die Idee des Gleichgewichtsmodells besteht nun darin, die sicheren Zustände als diejenigen zu definieren, in denen Verpflichtungen und ihre Beweise im Gleichgewicht sind. Es dürfen demnach keine zwei unbewiesenen Verpflichtungsänderungen aufeinander folgen, sondern es muss ein Beweis der ersteren dazwischen liegen. Die Erfüllung eigener Verpflichtungen muss sogar unmittelbar bewiesen werden. Dagegen werden die Anforderungen an die Verpflichtungsstruktur, etwa Verpflichtungen so zu organisieren, dass die Partner „Zug um Zug“ ihre Verpflichtungen eingehen und erfüllen, in das Vertrauensmodell verlegt. Auf diese

Weise verknüpft das Vertrauensmodell die (beweis)sicheren Zustände mit den erfolgsgekoppelten Verpflichtungsausdrücken und sorgt so für die Durchsetzung des Sicherheitsziels der Erfolgskopplung.

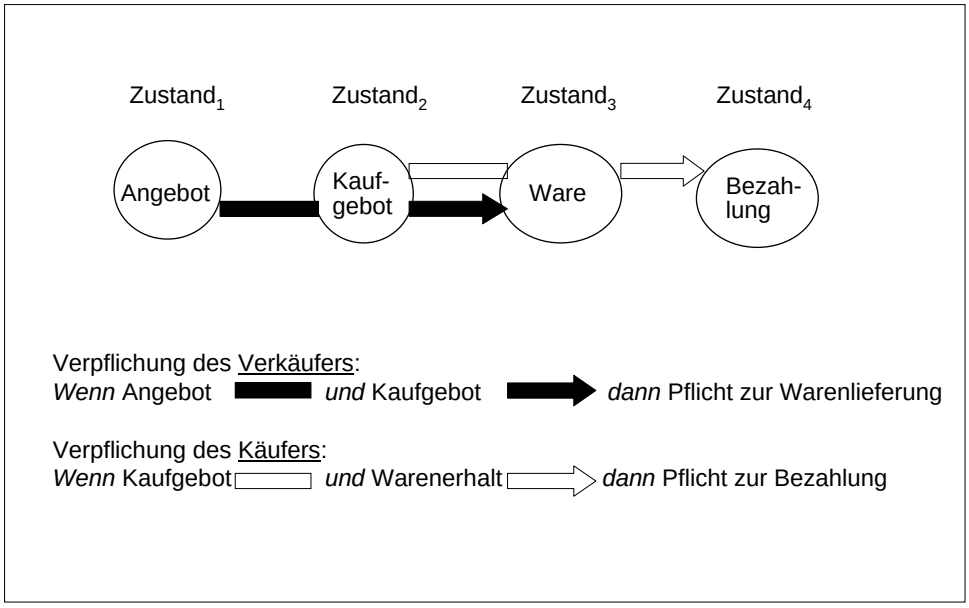

Abbildung 5: Der Sog ins Ziel durch aufeinander abgestimmte bedingte Verpflichtungsausdrücke

Die *Verpflichtungen* zum Handeln werden mit bedingten Verpflichtungsausdrücken beschrieben, die den Kooperationspartnern zugeordnet sind. In einer einfachen Verkaufskooperation gelten zum Beispiel die folgenden beiden bedingten Verpflichtungen. Für den Verkäufer gilt: „Wenn er eine Ware zu einem bestimmten Preis anbietet (Angebot) und er erhält ein zugehöriges Kaufversprechen (Kaufgebot), dann muss er die Ware liefern (Ware)." Für den Käufer gilt: „Wenn er ein Kaufversprechen abgibt (Kaufgebot) und die zugehörige Ware erhält (Ware), dann muss er den vereinbarten Preis bezahlen (Bezahlung)." Wenn nun beide Seiten ihren Verpflichtungen nachkommen, dann ziehen sie sich gewissermaßen gegenseitig Zug um Zug in das gemeinsame Kooperationsziel und beenden die Kooperation erfolgreich. Auch ein geregelter Abbruch ist akzeptabel: der Verkäufer bietet an, aber der Käufer gibt kein Kaufgebot, oder das Kaufgebot des Käufers entspricht nicht dem Angebot des Verkäufers. In diesem Falle erreicht keiner der beiden Kooperationspartner sein Ziel.

Bewertung:

Das Gleichgewichtsmodell beschreibt, wie sich ein Nutzer einer offenen Anwendungsumgebung (z.B. E-Commerce im Internet), in der er sich nicht auf das Wohlverhalten seiner Kooperationspartner verlassen kann, auf faire Austauschverfahren einlassen und dabei sicher sein kann, dass er sein Teilziel garantiert erreicht, wenn er seinen Kooperationspartner dessen Teilziel erreichen lässt. Es koppelt die Transaktionen der kooperierenden Partner derart aneinander, dass sie bei modellgetreuer Ausführung beide zum Erfolg kommen. In diesem Sinne ist es ein *erweitertes Transaktionsmodell*, man kann es ein *Erfolgskopplungsmodell* nennen.

Das Gleichgewichtsmodell ist ein semi-formales Modell mit formalen Ausdrücken des Kooperationsziels (Tausch der Waren), der Verpflichtungen, sowie der Regel zur Lieferung von Beweisen. Dass dadurch aber Fairness hergestellt wird, d.h. dass damit das Sicherheitsziel des Modells tatsächlich erreicht wird, ist formal nicht beweisbar, da die

signierten Aussagen außerhalb des Netzes als juristische Beweismittel genutzt werden. Das Modell setzt also in seinem Vertrauensmodell ein funktionierendes juristisches Umfeld voraus, in dem digitale Signaturen akzeptiert werden.

Die Abarbeitung von Verpflichtungsausdrücken, die Lieferung von Beweisen und die Warnung beim Ausbleiben von Beweisen sind informatorisch implementierbar. Der Aufwand ist aber insofern hoch, als die Akteure lokale Speicher für Beweismittel und Interpretationshilfen für ihre Bewertung brauchen. Dazu wird eine funktionierende *Public-Key*-Infrastruktur mit zugehörigen neutralen Entscheidungsinstanzen gebraucht, die im ganzen offenen Netz akzeptiert werden. Diese sind bis heute aber nicht etabliert. Statt dessen funktioniert das E-Commerce großer Firmen mit Privatkunden ohne Signaturen aufgrund von Kulanzregeln der Verkäufer gegenüber den Käufern und aufgrund von Markenvertrauen der Käufer gegenüber den Verkäufern. Im E-Commerce kleiner Firmen sind geschlossene Umgebungen wie E-Bay erfolgreich, die aufgrund ihrer zentralen Kontrolle ebenfalls auf Signaturen und Gleichgewichtsregeln verzichten können: Fehlverhalten wird hier durch Reputationsverlust bis hin zum Ausschluss sanktioniert.

Ausführung:

Das Modell definiert *Kooperationsziele, Verpflichtungen* und *Beweisregeln* und formuliert mit ihrer Hilfe das *Sicherheitsziel, die sicheren Zustände und die erlaubten Zustandsübergänge.* Das *Vertrauensmodell* schließlich beschreibt die Verpflichtungsstruktur und ihre Einbettung in die Wirklichkeit, mit der die sicheren Zustände die Erfolgskopplung in einer Kooperation absichern.

Eine Kooperation hat die Erfüllung von Verpflichtungen als *Kooperationsziel*. Das Kooperationsziel kann in Form von Einzelverpflichtungen, die den Kooperationspartnern zugeschrieben werden, in ihre Teilziele zerlegt werden. Zum Beispiel ist das Teilziel eines Verkäufers der Erhalt der Bezahlung. Das Teilziel des Käufers ist der Erhalt der Ware. Das Kooperationsziel der Verkaufskooperation ist die Zusammensetzung der Teilziele: „Der Verkäufer erhält die Bezahlung und der Käufer erhält die Ware“. Achtung, das Kooperationsziel (die Erfüllung von Verpflichtungen) ist streng zu trennen vom Sicherheitsziel einer Kooperation.

Das **Sicherheitsziel** des Gleichgewichtsmodells ist die Erfolgskopplung der Kooperationspartner: entweder jeder oder kein Partner erreicht sein Kooperationsziel. Es ist als *Erfolgskopplungsmodell* ein *erweitertes Transaktionsmodell*.

Subjekte R, S, … tauschen in Kooperationsumgebungen über offene Netze untereinander Nachrichten der Typen i, k, m, … aus. Jede Kooperation hat ein spezifiziertes Ziel der Art „S(i:R) und R(k:S)“, welches den Sachverhalt „S hat Nachricht vom Typ i von R erhalten und R hat Nachricht vom Typ k von S erhalten“ bezeichnet. Das Versenden einer Nachricht m von S an R wird mit einem Negativzeichen vor der Nachricht notiert: „S(-m:R)“ bedeutet: „S hat m an R geschickt“.

Eine *Verpflichtung* für S, eine Nachricht vom Typ m an R zu schicken, wird mit einem „V“ notiert: „V(S(-m:R))“. Bedingte Verpflichtungen können nun so ausgedrückt werden:

> „S(-i:R) *und* S(k:R) $\Rightarrow$ V(S(-m:R)“ ist so zu verstehen: „Wenn S eine Nachricht vom Typ i an R geschickt hat und wenn S eine Nachricht vom Typ k von R erhalten hat, dann ist S verpflichtet, eine Nachricht vom Typ m an S zu schicken“.

Die *einfache Verkaufskooperation*, in der das Prinzip „erst die Ware, dann das Geld“ gilt, ist durch das folgende Ziel und die beiden folgenden bedingten Verpflichtungsausdrücke charakterisiert:

Teilnehmer:	R=Käufer, S=Verkäufer
Nachrichtentypen:	Angebot, Kaufgebot, Ware, Bezahlung
Sicherheitsziel:	S(Bezahlung:R) ⇔ R(Ware:S)
Sicherheitsziel von S:	R(Ware:S) ⇒ S(Bezahlung:R)
Sicherheitsziel von R:	S(Bezahlung:R) ⇒ R(Ware:S)
Verpflichtungen:	
V(S):	S(-Angebot:R) *und* S(Kaufgebot:R) ⇒ V(S(-Ware:R))
V(R):	R(-Kaufgebot:S) *und* R(Ware:S) ⇒ V(R(-Bezahlung:S))

Tabelle 2: Ziel und Verpflichtungen bei der einfachen Verkaufskooperation

Zur Unterstützung der Verpflichtungen bieten sich die Kooperationspartner an, einander Beweismittel für ihre Verpflichtungen und die Erfüllung ihrer Verpflichtungen in die Hand zu geben. Der Beweis eines Nachrichtempfangs ist die Nachricht selbst. Damit der Sender sie nicht abstreiten kann, muss sie digital signiert sein. R(Angebot:S) bezeichnet das von S signierte Angebot in der Hand von R. Der Beweis einer Nachrichtensendung ist der Empfang einer zugehörigen Quittung, die zur Nicht-Abstreitbarkeit ebenfalls digital signiert sein muss. S(Quittung(Ware):R) bezeichnet die von R signierte Empfangsquittung der Warenlieferung von S in der Hand von S.

Die folgenden Regeln legen fest, wann ein Teilnehmer, der sich wohl verhält, einen Beweis über seine Verpflichtungen liefert, und wann ein Teilnehmer von seinem Partner einen Beweis fordert, bevor er die Kooperation seinerseits fortsetzt. Die Regeln lauten:

Beweisregel 1: Jeder Teilnehmer muss seinem Kooperationspartner (bzw. seinen Kooperationspartnern) für die Erfüllung der Voraussetzung eines Verpflichtungsausdrucks, der *ihm selbst* zugeordnet ist, ein Beweismittel liefern: „Ich muss dir ein Beweismittel liefern für jedes Ereignis, das zur Voraussetzung *meines* Verpflichtungsausdrucks gehört."

Beweisregel 2: Jeder Teilnehmer muss seinem Kooperationspartner (bzw. seinen Kooperationspartnern) für die Erfüllung einer Verpflichtung aus *dessen* Verpflichtungsausdruck ein Beweismittel liefern: „Ich muss dir ein Beweismittel liefern für jedes Ereignis, durch das *du deine* Verpflichtung erfüllst."

In der einfachen Verkaufskooperation greift die Regel 1 auf Seiten des Verkäufers S bei den Voraussetzungen S(-Angebot:R) und S(Kaufgebot:R) seines Verpflichtungsausdrucks V(S). Sobald eines dieser Ereignisse eintritt, muss er dem Käufer R darüber ein Beweismittel liefern: S(-Angebot:R) wird durch seine digitale Signatur des Angebots zum Beweismittel. Für S(Kaufgebot:R) gibt er mit einer signierten Quittung S(-Quittung(Kaufgebot):R) R ein Beweismittel in die Hand.

In der einfachen Verkaufskooperation greift die Regel 2 auf Seiten des Verkäufers S bei der zugehörigen Verpflichtung S(-Ware:R). Über seine Warenlieferung erwartet der Verkäufer einen Beweis vom Käufer R, die dieser durch eine digital signierte Empfangsbestätigung R(-Quittung(Ware):S) erbringt. Zur Vermeidung von quittungsunwilligen Partnern kann man die gesamte Kommunikation auch über neutrale Treuhänder abwickeln, die für die Empfangsbestätigungen sorgen.

Spezifikation eines sicheren Systemzustands: Ein sicherer Zustand eines Kooperationsteilnehmers im Sinne des Gleichgewichtsmodells ist ein Zustand, der zwei Bedingungen erfüllt: erstens kann ein Kooperationsteilnehmer die Verpflichtung seiner Partner wenigstens bis zum unmittelbar davorliegenden Zustand sowie die Freiheit eigener Verpflichtungen unmittelbar beweisen.

Erlaubte Zustandsübergänge: Jeder Verpflichtung folgt ihr Beweis an den Nutznießer der Verpflichtung (Beweisregel 1) spätestens im nächsten Schritt. Jeder Erfüllung einer Verpflichtung folgt ihr Beweis (Quittung) an den, der die Verpflichtung erfüllt hat (Beweisregel 2), und zwar im selben Schritt.

Sicherheitstheorem: Wenn ein Teilnehmer einer Kooperation erst dann den nächsten verbindlichen Schritt unternimmt, wenn er alle *bisherigen* Verpflichtungen des Partners bzw. seine aktuelle Freiheit von Verpflichtungen beweisen kann (erlaubter Zustandsübergang), dann hat er immer nur höchstens einen unbewiesenen Schritt in Bezug auf die Verpflichtung seines Partners unternommen und kann seine eigene Freiheit von Verpflichtungen jederzeit beweisen.

Der Beweis des Sicherheitstheorems ist trivial, da die Beweisregeln gerade so gemacht sind, dass ein sicherer Systemzustand einen unbewiesenen Schritt erlaubt (aber nicht mehr), der aber im nächsten Schritt ausgeglichen werden muss.

Das **Vertrauensmodell** verbindet die sicheren Systemzustände mit der „Erfolgskopplung“ einer Kooperation, indem es Voraussetzungen über die Wirkung von Beweisen formuliert, eine geeignete Systemimplementierung beschreibt und Ratschläge für die Kooperationsschritte erteilt. Ein Teil der Systemimplementierung ist die Formulierung der Verpflichtungsausdrücke. Diese selbst, sowie die Anforderung an ihre kooperationsgerechte Struktur, sind formalisierbar.

Wenn ein konkretes Kooperationssystem seine Verpflichtungsausdrücke zum Beispiel so strukturiert, dass die Verpflichtungen wechselseitig „Zug um Zug“ eingegangen und erfüllt werden, dann kann sich ein Teilnehmer davor schützen, dass sein Partner sein Kooperationsziel ohne ihn erreicht, indem er sich an die Regeln für erlaubte Zustandsübergänge hält. Das Vertrauensmodell rät einem Teilnehmer nun dazu, beim Ausbleiben eines erwarteten Beweises die Kooperation abzubrechen, bevor Schlimmeres passiert. Und wenn er dann seine Verpflichtung erfüllt, bevor sein Partner das getan hat (einer muss ja den ersten Schritt tun), dann verfügt er wenigstens über einen vollständigen Beweis darüber, dass sein Partner verpflichtet ist, seine Verpflichtung ebenfalls zu erfüllen. Das Vertrauensmodell setzt eine entsprechende Beweiswirkung voraus, so dass die Erfüllung einer nicht eingegangenen Verpflichtung im juristisch-sozialen Umfeld durchgesetzt werden kann.

Wenn sich die Partner zusätzlich an die Handlungsregel ihrer Verpflichtungsausdrücke halten, dann ziehen die Verpflichtungsausdrücke die Kooperationspartner Zug um Zug ins gemeinsame Kooperationsziel, das heißt, in diesem Fall kommen beide Partner zum Erfolg.

Um den Begriff des „Gleichgewichts“ zu verdeutlichen, demonstriert Tabelle 3 den Fortschritt einer Kooperation von oben nach unten über fünf Zustände, in dem die Entwicklung von Verpflichtungen durch zunehmende Beweise begleitet wird. In der linken Hälfte entwickeln sich die Verpflichtungszustände von oben nach unten, in der rechten Hälfte die Beweismittel. Am Anfang gibt es nur bedingte Verpflichtungen (links oben), also noch keine direkten Verpflichtungen und entsprechend noch keine Beweise (rechts oben). Mit der fortschreitenden Kooperation nehmen die Verpflichtungen an Gewicht zu (links: aus bedingten Verpflichtungen werden unbedingte Verpflichtungen), ebenso die Beweise (rechts untere Hälfte). Am Ende lösen sich die Verpflichtungen durch ihre Erfüllung wieder auf und übrig bleiben die Beweise in den Speichern der Teilnehmer für eventuelle spätere Reklamationen (letzte Zeile).

	Verpflichtungszustände für		*Beweise in Besitz von*	
	Verkäufer S	Käufer R	Verkäufer S	Käufer R
Vor Schritt 1: Vor Warenangebot	S(-Angebot:R) und S(Kaufgebot:R) ⇒ V(S(-Ware:R))	R(-Kaufgebot:S) und R(Ware:S) ⇒ V(R(-Bezahlung:S))	–	–
Zustand 1: Nach Warenangebot	S(Kaufgebot:R) ⇒ V(S(-Ware:R))	R(-Kaufgebot:S) und R(Ware:S) ⇒ V(R(-Bezahlung:S))	–	R(Angebot:S)
Zustand 2: Nach Kaufgebot	V(S(-Ware:R))	R(Ware:S) ⇒ V(R(-Bezahlung:S))	S(Kaufgebot:R)	R(Angebot:S), R(Quittung(Kaufgebot):S)
Zustand 3: Nach Warenlieferung	–	V(R(-Bezahlung:S))	S(Kaufgebot:R), S(Quittung(Ware):R)	R(Angebot:S), R(Quittung(Kaufgebot):S), R(Ware:S)
Zustand 4: Nach Bezahlung	–	–	S(Kaufgebot:R), S(Quittung(Ware):R), S(Bezahlung:R)	R(Angebot:S), R(Quittung(Kaufgebot):S), R(Ware:S), R(Quittung(Bezahlung):S)

Tabelle 3: Gleichgewicht aus Verpflichtungen und Beweisen in der einfachen Verkaufskooperation nach dem Prinzip „Erst die Ware – dann das Geld“

7. Was haben wir gelernt?

Modelle liefern vereinfachte Beschreibungen ihrer Gegenstände, die von unwichtigen Aspekten abstrahieren und dadurch wichtige Aspekte hervorheben. IT-Sicherheitsmodelle bezeichnen immer ihr Sicherheitsziel, das eine bestimmte Teilmenge von Sicherheitsanforderungen für eine typische Anwendungsumgebung repräsentiert. Sie definieren sichere Systemzustände und formulieren Regeln, wie man von sicheren Systemzuständen wieder in sichere Systemzustände gelangt. Ein Vertrauensmodell innerhalb eines IT-Sicherheitmodells nennt die Anforderungen an die Auslegung eines Systems und an seine Anwendungsumgebung, unter denen ein System dadurch sein Sicherheitsziel erreicht, dass es nach den Zustandsübergangsregeln in sicheren Zuständen verbleibt. Je formaler ein IT-Sicherheitsmodell ist, desto formaler wird die Wirkung der Regeln für das behauptete Sicherheitsziel bewiesen.

Die Aufgabe von IT-Sicherheitsmodellen ist die möglichst präzise Beschreibung von Sicherheitsanforderungen mit einem Hinweis darauf, welche Regeln zugehörige Sicherheitsmechanismen zu implementieren hätten. Die vier hier vorgestellten Modelle halten sich alle an dieses Grundmuster. Sie beschreiben unterschiedliche, typische Sicherheitsanforderungen, die „übergeordnete Sicherheitsziele“ genannt werden. Da die standardisierten Kriterien zur Bewertung der Sicherheit von IT-Produkten je nach Prüftiefe Sicherheitsmodelle in unterschiedlicher Detailgenauigkeit verlangen, werden zunehmend IT-Sicherheitsmodelle für anwendungstypische Schutzprofile gebraucht. Die hier vorgestellten Modelle können dafür ein hilfreiches Vorbild abgeben.

Literatur

Bell, D.E.; LaPadula, L.J. (1973): Secure Computer Systems: Mathematical Foundations, and A mathematical model. ESD-TR-73-278, MTR-2547, Vols 1&2. The MITRE Corporation, Bedford, MA, Nov 1973.

Bell, D.E.; LaPadula, L.J. (1975): Computer Security Model: Unified Exposition and Multics Interpretation. ESD-TR-75-306, The MITRE Corporation, Bedford, MA, June 1975.

Brewer, D.F.C.; Nash, M.J. (1989): The Chinese Wall Security Policy. Proceedings of the 1989 IEEE Symposium on Security and Privacy, Oakland, CA. Computer Society Press of the IEEE, Washington DC, 206-214, 1989.

CC (2006): Common Criteria for Information Technology Security Evaluation, Version 3.1 Revision 1, September 2006, http://www.commoncriteriaportal.org/ [29.12.2007].

Clark, D.; and Wilson, D. (1987): A Comparison of Commercial and Military Security Policies. Proceedings of the 1987 IEEE Symposium on Security and Privacy, Oakland, CA. Computer Society Press of the IEEE, Washington DC, 184-194, 1987.

Clark, D.; and Wilson, D. (1988): Evaluation of a Model for Computer Integrity. Paper for presentation at the 11th National Computer Security Conference, Baltimore, MY, Oct 17-20 1988. Copyright Ernst&Whinney, 1988, 13 pages.

Cohen, Fred (1987): Computer Viruses. Proceedings of IEEE Computer and Security 6, 1987, 22-35.

Eckert, Claudia (2006): IT-Sicherheit. Konzepte, Verfahren, Protokolle. Studienausgabe. Oldenbourg Verlag, München, 2006, Kap. 4.

Grimm, Rüdiger (1993): Non-repudiation in Open Telecooperation. Paper for presentation at the 16th National Computer Security Conference (NCSC). Baltimore, MY, Sep 20-23, 1993.

Grimm, Rüdiger (1994): Sicherheit für offene Kommunikation – Verbindliche Telekooperation. B.I. Wissenschaftsverlag, Mannheim, 1994.

von Neumann, John (1945): First Draft of a Report on the EDVAC. Contract No. W-670-ORD-492, Moore School of Electrical Engineering, Univ. of Penn., Philadelphia, 30. June 1945. Teilw. nachgedruckt in Randell, Brian: Origins of Digital Computers: Selected Papers, Springer-Verlag, Berlin Heidelberg, 1982, pp. 383-392.

Terry, P.; and Wiseman, Simon R. (1989): A New Security Model. Proc. IEEE Symposium on Research in Security and Privacy, 215-228, 1989.

Bisher erschienen

Arbeitsberichte aus dem Fachbereich Informatik
(http://www.uni-koblenz.de/fb4/publikationen/arbeitsberichte)

Rüdiger Grimm: IT-Sicherheitsmodelle, Arbeitsberichte aus dem Fachbereich Informatik 3/2008

Rüdiger Grimm, Helge Hundacker, Anastasia Meletiadou: Anwendungsbeispiele für Kryptographie (erweiterte Version), Arbeitsberichte aus dem Fachbereich Informatik 2/2008

Markus Maron, Kevin Read, Michael Schulze: CAMPUS NEWS – Artificial Intelligence Methods Combined for an Intelligent Information Network, Arbeitsberichte aus dem Fachbereich Informatik 1/2008

Lutz Priese,Frank Schmitt, Patrick Sturm, Haojun Wang: BMBF-Verbundprojekt 3D-RETISEG Abschlussbericht des Labors Bilderkennen der Universität Koblenz-Landau, Arbeitsberichte aus dem Fachbereich Informatik 26/2007

Stephan Philippi, Alexander Pinl: Proceedings 14. Workshop 20.-21. September 2007 Algorithmen und Werkzeuge für Petrinetze, Arbeitsberichte aus dem Fachbereich Informatik 25/2007

Ulrich Furbach, Markus Maron, Kevin Read: CAMPUS NEWS – an Intelligent Bluetooth-based Mobile Information Network, Arbeitsberichte aus dem Fachbereich Informatik 24/2007

Ulrich Furbach, Markus Maron, Kevin Read: CAMPUS NEWS - an Information Network for Pervasive Universities, Arbeitsberichte aus dem Fachbereich Informatik 23/2007

Lutz Priese: Finite Automata on Unranked and Unordered DAGs Extented Version, Arbeitsberichte aus dem Fachbereich Informatik 22/2007

Mario Schaarschmidt, Harald F.O. von Kortzfleisch: Modularität als alternative Technologie- und Innovationsstrategie, Arbeitsberichte aus dem Fachbereich Informatik 21/2007

Kurt Lautenbach, Alexander Pinl: Probability Propagation Nets, Arbeitsberichte aus dem Fachbereich Informatik 20/2007

Rüdiger Grimm, Farid Mehr, Anastasia Meletiadou, Daniel Pähler, Ilka Uerz: SOA-Security, Arbeitsberichte aus dem Fachbereich Informatik 19/2007

Christoph Wernhard: Tableaux Between Proving, Projection and Compilation, Arbeitsberichte aus dem Fachbereich Informatik 18/2007

Ulrich Furbach, Claudia Obermaier: Knowledge Compilation for Description Logics, Arbeitsberichte aus dem Fachbereich Informatik 17/2007

Fernando Silva Parreiras, Steffen Staab, Andreas Winter: TwoUse: Integrating UML Models and OWL Ontologies, Arbeitsberichte aus dem Fachbereich Informatik 16/2007

Rüdiger Grimm, Anastasia Meletiadou: Rollenbasierte Zugriffskontrolle (RBAC) im Gesundheitswesen, Arbeitsberichte aud dem Fachbereich Informatik 15/2007

Ulrich Furbach, Jan Murray, Falk Schmidsberger, Frieder Stolzenburg: Hybrid Multiagent Systems with Timed Synchronization-Specification and Model Checking, Arbeitsberichte aus dem Fachbereich Informatik 14/2007

Björn Pelzer, Christoph Wernhard: System Description:“E-KRHyper“, Arbeitsberichte aus dem Fachbereich Informatik, 13/2007

Ulrich Furbach, Peter Baumgartner, Björn Pelzer: Hyper Tableaux with Equality, Arbeitsberichte aus dem Fachbereich Informatik, 12/2007

Ulrich Furbach, Markus Maron, Kevin Read: Location based Informationsystems, Arbeitsberichte aus dem Fachbereich Informatik, 11/2007

Philipp Schaer, Marco Thum: State-of-the-Art: Interaktion in erweiterten Realitäten, Arbeitsberichte aus dem Fachbereich Informatik, 10/2007

Ulrich Furbach, Claudia Obermaier: Applications of Automated Reasoning, Arbeitsberichte aus dem Fachbereich Informatik, 9/2007

Jürgen Ebert, Kerstin Falkowski: A First Proposal for an Overall Structure of an Enhanced Reality Framework, Arbeitsberichte aus dem Fachbereich Informatik, 8/2007

Lutz Priese, Frank Schmitt, Paul Lemke: Automatische See-Through Kalibrierung, Arbeitsberichte aus dem Fachbereich Informatik, 7/2007

Rüdiger Grimm, Robert Krimmer, Nils Meißner, Kai Reinhard, Melanie Volkamer, Marcel Weinand, Jörg Helbach: Security Requirements for Non-political Internet Voting, Arbeitsberichte aus dem Fachbereich Informatik, 6/2007

Daniel Bildhauer, Volker Riediger, Hannes Schwarz, Sascha Strauß, „grUML – Eine UML-basierte Modellierungssprache für T-Graphen“, Arbeitsberichte aus dem Fachbereich Informatik, 5/2007

Richard Arndt, Steffen Staab, Raphaël Troncy, Lynda Hardman: Adding Formal Semantics to MPEG-7: Designing a Well Founded Multimedia Ontology for the Web, Arbeitsberichte aus dem Fachbereich Informatik, 4/2007

Simon Schenk, Steffen Staab: Networked RDF Graphs, Arbeitsberichte aus dem Fachbereich Informatik, 3/2007

Rüdiger Grimm, Helge Hundacker, Anastasia Meletiadou: Anwendungsbeispiele für Kryptographie, Arbeitsberichte aus dem Fachbereich Informatik, 2/2007

Anastasia Meletiadou, J. Felix Hampe: Begriffsbestimmung und erwartete Trends im IT-Risk-Management, Arbeitsberichte aus dem Fachbereich Informatik, 1/2007

„Gelbe Reihe“

(http://www.uni-koblenz.de/fb4/publikationen/gelbereihe)

Lutz Priese: Some Examples of Semi-rational and Non-semi-rational DAG Languages. Extended Version, Fachberichte Informatik 3-2006

Kurt Lautenbach, Stephan Philippi, and Alexander Pinl: Bayesian Networks and Petri Nets, Fachberichte Informatik 2-2006

Rainer Gimnich and Andreas Winter: Workshop Software-Reengineering und Services, Fachberichte Informatik 1-2006

Kurt Lautenbach and Alexander Pinl: Probability Propagation in Petri Nets, Fachberichte Informatik 16-2005

Rainer Gimnich, Uwe Kaiser, and Andreas Winter: 2. Workshop "Reengineering Prozesse" – Software Migration, Fachberichte Informatik 15-2005

Jan Murray, Frieder Stolzenburg, and Toshiaki Arai: Hybrid State Machines with Timed Synchronization for Multi-Robot System Specification, Fachberichte Informatik 14-2005

Reinhold Letz: FTP 2005 – Fifth International Workshop on First-Order Theorem Proving, Fachberichte Informatik 13-2005

Bernhard Beckert: TABLEAUX 2005 – Position Papers and Tutorial Descriptions, Fachberichte Informatik 12-2005

Dietrich Paulus and Detlev Droege: Mixed-reality as a challenge to image understanding and artificial intelligence, Fachberichte Informatik 11-2005

Jürgen Sauer: 19. Workshop Planen, Scheduling und Konfigurieren / Entwerfen, Fachberichte Informatik 10-2005

Pascal Hitzler, Carsten Lutz, and Gerd Stumme: Foundational Aspects of Ontologies, Fachberichte Informatik 9-2005

Joachim Baumeister and Dietmar Seipel: Knowledge Engineering and Software Engineering, Fachberichte Informatik 8-2005

Benno Stein and Sven Meier zu Eißen: Proceedings of the Second International Workshop on Text-Based Information Retrieval, Fachberichte Informatik 7-2005

Andreas Winter and Jürgen Ebert: Metamodel-driven Service Interoperability, Fachberichte Informatik 6-2005

Joschka Boedecker, Norbert Michael Mayer, Masaki Ogino, Rodrigo da Silva Guerra, Masaaki Kikuchi, and Minoru Asada: Getting closer: How Simulation and Humanoid League can benefit from each other, Fachberichte Informatik 5-2005

Torsten Gipp and Jürgen Ebert: Web Engineering does profit from a Functional Approach, Fachberichte Informatik 4-2005

Oliver Obst, Anita Maas, and Joschka Boedecker: HTN Planning for Flexible Coordination Of Multiagent Team Behavior, Fachberichte Informatik 3-2005

Andreas von Hessling, Thomas Kleemann, and Alex Sinner: Semantic User Profiles and their Applications in a Mobile Environment, Fachberichte Informatik 2-2005

Heni Ben Amor and Achim Rettinger: Intelligent Exploration for Genetic Algorithms – Using Self-Organizing Maps in Evolutionary Computation, Fachberichte Informatik 1-2005