This is an author-created version of the following work: P. Ochsenschläger, R. Rieke and Z. Velikova (2008). Die elektronische Krankenakte - Eine Sicherheitsstrategie. Published in: DACH Security 2008 - Bestandsaufnahme, Konzepte, Anwendungen, Perspektiven. (ISBN: 978-3-00-024632-6), pages 90-100

Die elektronische Krankenakte - Eine Sicherheitsstrategie

Peter Ochsenschläger[1] · Roland Rieke[1] · Zaharina Velikova[1]

[1]Fraunhofer-Institut für Sichere Informationstechnologie
{peter.ochsenschlaeger,roland.rieke,zaharina.velikova}@sit.fraunhofer.de

Zusammenfassung

Diese Arbeit stellt ein Organisationsstruktur-basiertes Berechtigungskonzept am Beispiel der elektronischen Krankenakte vor. Ein ausführbares Modell der Sicherheitsstrategie erlaubt die vollständige Analyse des Systemverhaltens. Durch eigenschaftserhaltende Abstraktionen wird daraus eine kompakte Visualisierung von spezifischen Aspekten dieses Verhaltens möglich. Wegen der Kompaktheit der Darstellung lassen sich daraus die gewünschten Systemeigenschaften unmittelbar ablesen. Da die gewählten Abstraktionen eigenschaftserhaltend sind, folgen daraus nicht nur die erforderlichen Sicherheitseigenschaften sondern auch die Funktionssicherheit eines solchen Berechtigungskonzeptes trotz der gebotenen Datensparsamkeit.

1 Einführung

Welche Zugriffe auf Daten in einem Krankenhaus zulässig sind und wie weitgehend die Daten geschützt werden müssen, wird durch europäische und deutsche Datenschutzgesetze und die Landeskrankenhausgesetze der einzelnen Bundesländer geregelt. Im Spannungsfeld zwischen dem Konzept der Datensparsamkeit in Bezug auf personenbezogene Daten, das in diesem Kontext aufgrund der hohen Sensibilität der medizinischen Daten besonders durch die ärztliche Schweigepflicht (das "Patientengeheimnis") und das "Informationelle Selbstbestimmungsrecht des Patienten" gestützt wird, einerseits, und dem Prinzip der Erforderlichkeit ("need to know") sowie der einem Arzt obliegenden Dokumentationspflicht andererseits, muss, basierend auf der Organisationsstruktur des Krankenhauses, ein differenziertes Berechtigungskonzept erarbeitet und durchgesetzt werden.

Ziel dieser Arbeit ist es, Sicherheits- und "Funktionseigenschaften" (Lebendigkeitseigenschaften) eines durch ein solches Berechtigungskonzept gesteuerten Systems am Beispiel eines Krankenhauses formal zu analysieren, um einerseits die Einhaltung der geforderten Datenschutzziele und andererseits die Verfügbarkeit der erforderlichen Daten für die Berechtigten nachzuweisen. Zur Umsetzung des Berechtigungskonzeptes wurde zunächst ein Konzept für eine Zugriffskontrolle basierend auf dem Or-BAC (Organization Based Access Control) Modell [ABB+03] erstellt. Darauf aufbauend wurden formale Methoden der Spezifikation und Verifikation von kooperierenden Systemen basierend auf den Konzepten asynchroner Produktautomaten (APA) und schlichter Homomorphismen, die bei Fraunhofer SIT entwickelt wurden [ORR00a] , angewendet, um die erforderlichen Eigenschaften nachzuweisen.

Ausgehend von einer Beschreibung der Organisationsstruktur und der Abläufe in einem Krankenhaus sowie der rechtlichen Anforderungen, wird mittels Or-BAC ein Zugriffskontrollmodell

für ein Beispielkrankenhaus erstellt. Dabei fließen neben den datenschutzrechtlichen Bestimmungen auch organisatorische und strukturelle Rahmenbedingungen mit ein. Darauf aufbauend wird das Zugriffskontrollregelwerk und der Workflow im Krankenhaus mittels kommunizierender Automaten (asynchrone Produktautomaten) formal modelliert und aus diesem operationalen Modell mittels des SH Verification Tool [ORR00b] das dynamische Systemverhalten in Form eines Erreichbarkeitsgraphen berechnet. Dieser bildet den Ausgangspunkt für die nun folgende Verifikation einiger ausgewählter Systemeigenschaften:

- Die korrekte Reihenfolge bestimmter Aktionen.
- Welche Aktionen von welchem Mitarbeiter auf welchem Dokument durchgeführt werden können.
- Die Durchsetzung unterschiedlicher Rechte für entsprechende Fachabteilungen.
- Die Möglichkeit bestimmter Aktionen wie z.B. der Erzeugung notwendiger Dokumente (Lebendigkeiteigenschaft).
- Sicherheitseigenschaften wie z.B. dass ein Dokument nicht gelöscht werden kann oder dass ein unberechtigter Mitarbeiter ein Dokument nicht lesen kann.

Bezüglich aller Punkte werden für die Spezifikation die gewünschten Eigenschaften nachgewiesen.

Der Beitrag ist wie folgt strukturiert. Kapitel 2 beschreibt ein konkretes Organisationsstrukturbasiertes Berechtigungskonzept am Beispiel einer elektronischen Krankenakte. Ein Zugriffskontrollmodell und ein Modell der möglichen Vorgänge (Workflow) werden vorgestellt. In Kapitel 3 wird die Verifikation von Sicherheits- und Lebendigkeiteigenschaften an diesem Modell exemplarisch dargestellt. Eine kompakte Visualisierung von spezifischen Aspekten dieses Verhaltens ermöglicht es dabei, die gewünschten Systemeigenschaften unmittelbar abzulesen. Im Ausblick weisen wir dann auf weiterführende Arbeiten hin.

2 Modellierung einer Sicherheitsstrategie

2.1 Die elektronische Krankenakte

Die Krankenakte umfasst alle Daten über einen Patienten, die im Zusammenhang mit der medizinischen Versorgung erhoben und erstellt werden. Man kann hier zwischen personenbezogenen Dokumenten einerseits und behandlungsfallbezogenen Dokumenten andererseits unterscheiden. Zusätzlich können jeweils die Dokumente in administrative Dokumente und in medizinische Dokumente unterschieden werden. Abbildung 1 zeigt die weitere Untergliederung dieser Dokumente. *Aufenthaltsdokumente* beispielsweise geben Auskunft über die aktuellen Aufenthaltsorte von Patienten innerhalb des Krankenhauses wie Fachabteilung, Station und Zimmernummer.

2.2 Zugriff auf die elektronische Krankenakte

Der Zugriff auf Daten einer elektronischen Krankenakte darf nur unter der Wahrung der ärztlichen Schweigepflicht und unter der Berücksichtigung des informationellen Selbstbestimmungsrechtes des Patienten erfolgen. Die ärztliche Schweigepflicht wird auch Arztgeheimnis bzw. richtiger Patientengeheimnis genannt. Um der ärztlichen Schweigepflicht Genüge zu tun, reicht es nicht aus, sämtliche Zugriffe auf die elektronische Krankenakte zu protokollieren. Durch eine Protokollierung können zwar im Nachhinein unbefugte Zugriffe aufgedeckt

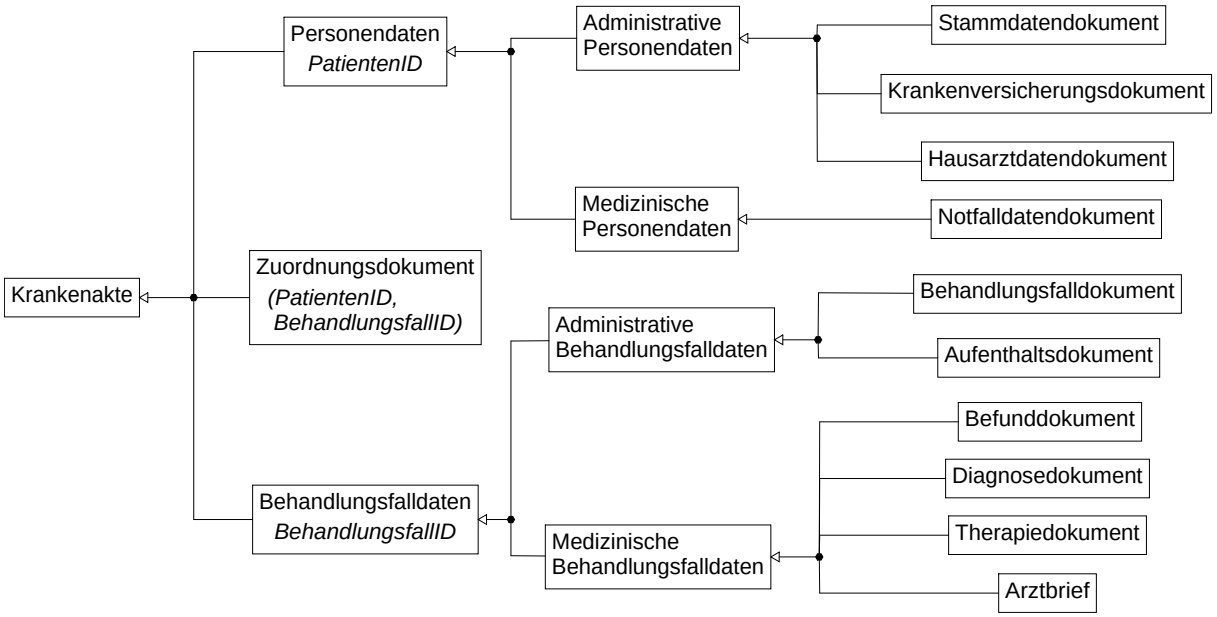

Abb. 1: Klassendiagramm zur elektronischen Krankenakte

werden, aber nicht unbefugte Zugriffe verhindert werden. “Es muss daher ein differenziertes Berechtigungskonzept erarbeitet werden, das basierend auf der Organisationsstruktur des Krankenhauses für alle Benutzer die zur Aufgabenstellung benötigten Rechte festlegt.”[fdD]

Ein differenziertes Berechtigungskonzept umfasst mehrere Dimensionen. Zum einen gibt es die Personen, welche auf die Daten von elektronischen Krankenakten zugreifen wollen. Es ist sinnvoll, die Personen mittels *Rollen*, die sie im Krankenhaus spielen, zusammenzufassen, damit die Komplexität reduziert wird. Zum anderen gibt es verschiedene *Rechte*, wie z.B. *anzeigen* und *bearbeiten*, die vergeben werden können. Des Weiteren gibt es die einzelnen Dokumente der elektronischen Krankenakte, auf die sich die Rechte beziehen. Hier ist es zweckmäßig, Dokumente nach ihrer Art zusammenzufassen, damit sich Rechte beispielsweise auf Befunde beziehen können. Für ein Berechtigungskonzept müssen die einzelnen Dimensionen einander zugeordnet werden [fdD].

Eine ausführliche Darstellung und Bewertung der relevanten rechtlichen Rahmenbedingungen im Rahmen der Gesundheitstelematik findet sich beispielsweise im übergreifenden Sicherheitskonzept für Umsetzung und Betrieb elektronischer Fallakten [Cau08].

Kernpunkt unseres Beitrags ist die Simulation und folgende Abstraktion eines Systems mit einem Organisationsstruktur-basierten Berechtigungskonzept. Die elektronische Krankenakte dient hier nur als ein Beispiel eines solchen Systems.

2.3 Or-BAC Modell eines Krankenhauses

Systeme, bei denen der Zugriff von Personen und Prozessen auf gemeinsame Ressourcen in einer differenzierten Form stattfinden soll, benötigen wohldefinierte Beschreibungen, welche Zugriffe erlaubt sind und welche nicht. Dazu ist wiederum erst einmal ein Beschreibungsmodell nötig, das einen Rahmen liefert, mit dessen Hilfe ebensolche Beschreibungen formuliert werden können. Solche Beschreibungsmodelle sind Zugriffskontrollmodelle. Mit einem Zugriffskontrollmodell kann ein Umfeld und eine darin geltende Zugriffskontrollstrategie formal

beschrieben werden. Diese besteht aus einem Regelwerk von Erlaubnissen, kann aber auch je nach Modell Verbote und Verpflichtungen umfassen. *Organization Based Access Control* (Or-BAC) ist ein solches Zugriffskontrollmodell [ABB+03] .

Das zentrale Konzept im Or-BAC Modell ist das Konzept der *Organisation*. Die Zugriffskontrollregeln sind in Or-BAC immer an Organisationen bzw. Organisationseinheiten gebunden. In unserem Modell verwenden wir die Organisationseinheiten *Patientenverwaltung*, *Chirurgie* und *Radiologie*.

Neben dem aus RBAC bekannten *Rollen*-Konzept für *Subjekte* gibt es in Or-BAC entsprechende Konzepte für *Aktionen* und *Objekte*, bei denen die Aktionen bzw. Objekte in einem allgemeineren Zusammenhang betrachtet werden. Aktionen können als zu Tätigkeiten gehörend angesehen werden und Objekte können unter Sichten verwendet werden. Die neu eingeführten Konzepte heißen dementsprechend *Tätigkeit* und *Sicht*.

In unserem Modell eines Krankenhauses verwenden wir die Rollen *Verwaltungsmitarbeiter*, *Arzt* und *Pflegekraft*. Mittels der Relation *Empower* wird festgelegt, in welcher Einrichtung des Beispielkrankenhauses welcher Angestellte welche Rolle spielt. Beispielsweise bedeutet *Empower(Patientenverwaltung, Klein, Verwaltungsmitarbeiter)*, dass Herr *Klein* ermächtigt ist in der Organisationseinheit *Patientenverwaltung* die Rolle *Verwaltungsmitarbeiter* zu spielen. Wir spezifizieren weiterhin:

Empower(Chirurgie, Brinkmann, Arzt),
Empower(Chirurgie, Mueller, Pflegekraft),
Empower(Radiologie, Schmidt, Arzt).

Weiterhin verwenden wir 8 Objekte (*Stammdatendokument_1*, *Befunddokument_2*, *Aufenthaltsdokument_1*, ...) die in 6 verschiedenen Sichten (*Interner_Befund*, *Aufenthalt*, ...) verwendet werden können. Mittels der Relation *Use* wird festgelegt, in welcher Einrichtung des Beispielkrankenhauses welches Dokument unter welcher Sicht verwendet wird. Wir spezifizieren:

Use(Patientenverwaltung, Aufenthaltsdokument_1, Aufenthalt),
Use(Chirurgie, Aufenthaltsdokument_1, Aufenthalt),
Use(Radiologie, Aufenthaltsdokument_1, Aufenthalt).

Mittels der Relation *Consider* wird festgelegt, in welcher Einrichtung des Beispielkrankenhauses welche Aktion zu welcher Tätigkeit gehört. Wir verwenden im Modell 21 Aktionen (*Befunddokument_sperren*, *Aufenthaltsdokument_bearbeiten*, ...) im Rahmen von 21 Tätigkeiten (*Befund_sperren*, *Aufenthalt_aendern*, ...). Dabei wurde jeder Tätigkeit genau eine Aktion zugeordnet.

Consider(Chirurgie, Aufenthaltsdokument_bearbeiten, Aufenthalt_aendern)
Consider(Radiologie, Aufenthaltsdokument_bearbeiten, Aufenthalt_aendern)

Zusätzlich ist es in Or-BAC möglich, *Kontexte* zu definieren. Kontexte können zwischen Subjekten, Aktionen und Objekten bestehen und beschreiben gewisse Umstände, die erfüllt sein müssen, damit eine *Erlaubnis* erteilt wird. Mittels der Relation *Define* wird festgelegt, in welcher Einrichtung des Beispielkrankenhauses welcher Kontext zwischen welchem Subjekt, welcher Aktion und welchem Objekt besteht. Im Modell verwenden wir 34 unterschiedliche Kontexte (*Befunddokument_freigegeben*, *Ort_stimmt_ueberein*, ...). Wobei *Ort_stimmt_ueberein* bedeutet, dass die in einem gegebenen Aufenthaltsdokument enthaltene Organisation mit einer

gegebenen Organisation übereinstimmt.

In Or-BAC werden die Zugriffskontrollregeln mittels einer Menge von Erlaubnissen beschrieben. Mittels der Relation *Permission* wird festgelegt, in welcher Einrichtung des Beispielkrankenhauses welche Rolle welche Tätigkeit auf welche Sicht in welchem Kontext ausüben darf. Beispielsweise kann durch eine Permission der folgende Sachverhalt ausgedrückt werden: "Wenn ein Patient in einer bestimmten Fachabteilung ist, dann können dortige Ärzte sich alle Befunde von dieser Fachabteilung anzeigen lassen, die den Patienten betreffen."

Daneben gibt es in Or-BAC auch die Möglichkeit Verbote und Verpflichtungen zu modellieren, dies wird in unserem Beispiel aber nicht genutzt.

2.4 Relevante Aktionen im Beispielkrankenhaus

Ein asynchroner Produktautomat(APA) modelliert nun die möglichen Vorgänge (Workflow) im Beispielkrankenhaus [Bre07] wie folgt:

In jedem Systemzustand des Beispielkrankenhauses versucht ein zufällig gewähltes Subjekt eine zufällig gewählte Aktion auf einem zufällig gewählten Objekt auszuführen. Die Entscheidung darüber, ob ein konkretes Subjekt eine konkrete Aktion auf einem konkreten Objekt ausführen kann, wird mittels der Or-BAC basierten Zugriffskontrolle getroffen.

Bei der Durchführung einer Aktion im Modell wird zwischen unterschiedlichen Typen von Aktionen unterschieden:

- Falls eine Aktion das Erzeugen oder Löschen eines Dokumentes beinhaltet wie z.B. die Aktion *Aufenthaltsdokument_erzeugen*, so wird bei der Duchführung dieser Aktion im Modell ein solches Dokument erzeugt und die entsprechende Dokumentenliste (im Beispiel die *Aufenthaltsdokumentliste*) entsprechend modifiziert. Im Fall der Aktion *Aufenthaltsdokument_erzeugen* wird außerdem die zuständige Fachabteilung (im Beispielszenario *Radiologie* oder *Chirurgie*) in das neu erzeugte Aufenthaltsdokument eingetragen.
- Die Aktion *Aufenthaltsdokument_bearbeiten* bewirkt im Modell, daß, falls sich der im Aufenthaltsdokument eingetragene Aufenthaltsort durch die Bearbeitung ändert, der neu zuzuweisende Aufenthaltsort eingesetzt wird.
- In einer Fachabteilung können dortige Ärzte alle Befunde freigeben und sperren, die dieser Fachabteilung zugeordnet sind. Für Befunde gibt es daher zusätzliche Aktionen zum Freigeben (*Befunddokument_freigeben*) und Sperren (*Befunddokument_sperren*) von Befunddokumenten. Die Durchführung einer solchen Aktion wird im Modell durch einen entsprechenden Eintrag im jeweiligen Befunddokument markiert.

Damit kann nun untersucht werden, ob das Systemverhalten verschiedene Eigenschaften erfüllt, insbesondere Eigenschaften im Hinblick auf die Sicherheitsstrategie, die im Beispielkrankenhaus vorherrschen soll.

3 Verifikation von Eigenschaften

Mit Hilfe des SH Verification Tools lässt sich das Systemverhalten des Beispielkrankenhauses in Form eines Erreichbarkeitsgraphen berechnen. Jeder Knoten des Erreichbarkeitsgraphen entspricht einem Zustand, in dem sich das Beispielkrankenhaus befinden kann. Jede Kante entspricht einer Aktion, die im Beispielkrankenhaus stattfinden kann.

Die Kanten des Erreichbarkeitsgraphen lassen sich durch Zustandsübergänge $(p, (e, i), q)$ darstellen, wobei e der gerade aktive *Elementarautomat* ist und i die konkrete *Interpretation* der Variablen dieses Automaten. Die Menge aller solcher Kanten bezeichnen wir mit Ψ. Eine formale Definition dazu findet sich in [OR07].

Das Verhalten eines APA wird generell nicht nur durch seinen Erreichbarkeitsgraphen, sondern genauso durch die Menge aller möglichen Folgen von Zustandsübergängen, ausgehend vom Anfangszustand q_0, beschrieben.
Die Folge $(q_0, (e_1, i_1), q_1)\ (q_1, (e_2, i_2), q_2)\ (q_2, (e_3, i_3), q_3) \ldots (q_{n-1}, (e_n, i_n), q_n)$ mit $e_k \in E$ (E ist die Menge aller Elementarautomaten) und $i_k \in \Phi_{e_k}$ (Φ_{e_k} ist die Menge aller möglichen Interpretationen des Elementarautomaten e_k) stellt eine mögliche Folge von Aktionen eines APA dar.

Die Menge $L \subset \Psi^*$ aller Aktionsfolgen mit Anfangszustand q_0 einschließlich der leeren Folge ϵ heißt die *Aktionssprache* des entsprechenden APA.

Die Berechnung ergibt für das gewählte Szenario einen Erreichbarkeitsgraphen mit 2433 Knoten und 44908 Kanten.

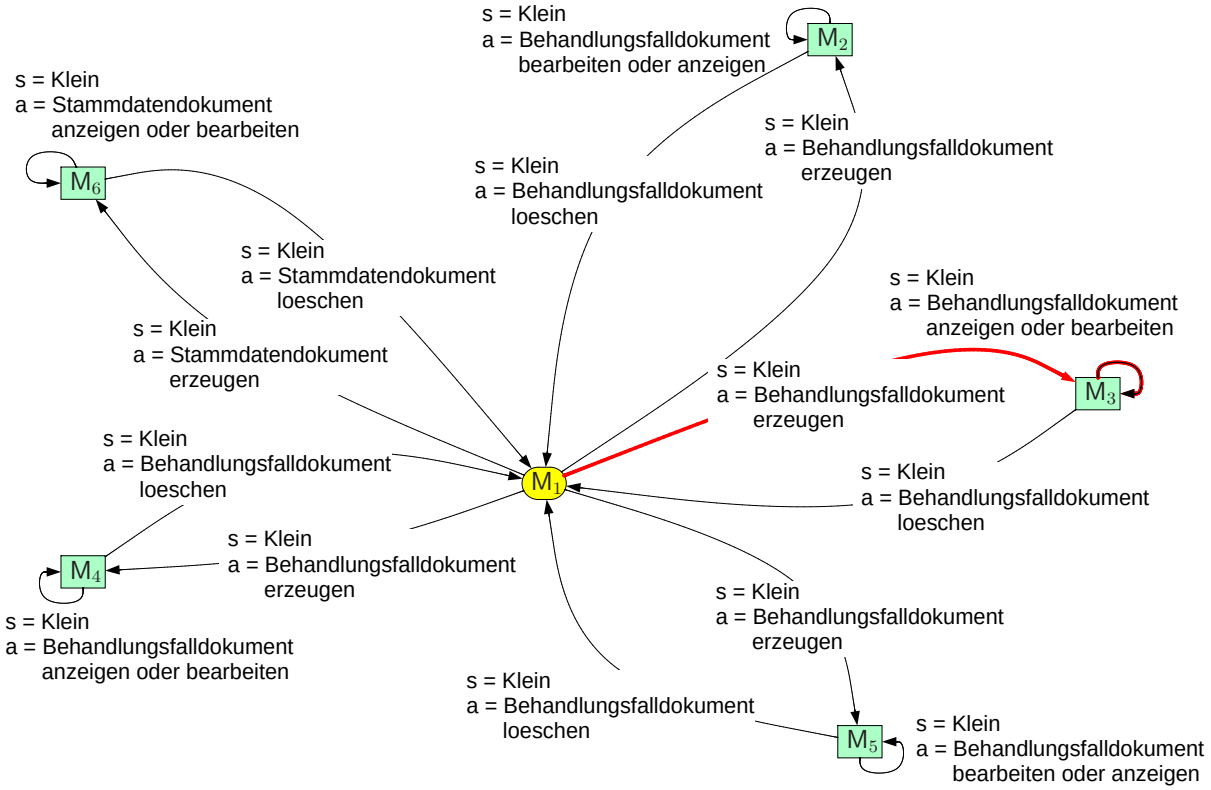

Abb. 2: Ausschnitt aus dem Erreichbarkeitsgraphen

Die Abbildung 2 zeigt die vom Anfangszustand M_1 *in einem Schritt* erreichbaren Folgezustände dieses Erreichbarkeitsgraphen. Ein Zustandsübergang vom Anfangszustand M_1 ist beispielsweise

$$(M_1, (T, (s = Klein, a = Behandlungsfalldokument\ erzeugen)), M_3) \qquad (1)$$

Hierbei wurden die nicht relevanten Interpretationen weggelassen. Dadurch sehen z.B. die Beschriftungen der Kanten $M_1 \to M_2$ und $M_1 \to M_3$ gleich aus. Die erzeugten Objekte sind aber unterschiedlich, so dass hier auch unterschiedliche Zustände erreicht werden.

Da der im Beispiel verwendete APA aus nur einem Elementarautomaten T besteht, haben wir ihn in den Kantenanschriften der Abbildung weggelassen.

Wenn man nun z.B. vom Zustand M_3 um einen Zustandsübergang weitergeht, so erreicht man mittels

$$(M_3, (T, (s = Klein, a = Aufenthaltsdokument\ erzeugen, org = Chirurgie)), M_{11}) \quad (2)$$

den Zustand M_{11}. Abbildung 3 stellt einen entsprechenden Ausschnitt aus dem Erreichbarkeitsgraphen dazu dar. Die Beschriftung der Kanten die sich nicht auf das Objekt *Aufenthaltsdokument_1* beziehen (gestrichelt dargestellt) wurde hier weggelassen. Von M_3 und M_{11} verzweigen wie auch von M_1 noch viele andere Pfade, die hier ebenfalls nicht eingezeichnet sind.

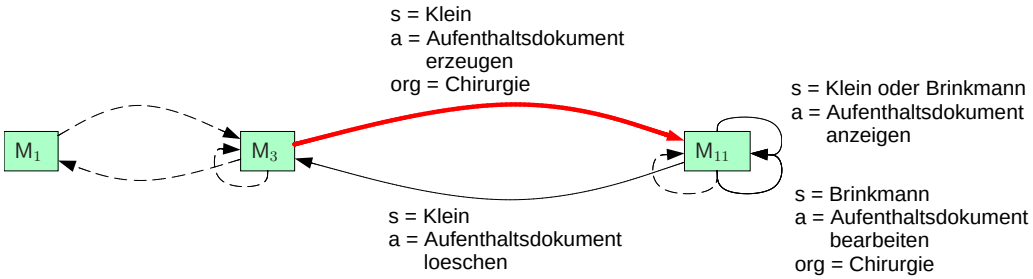

Abb. 3: Ein Anfangspfad im Erreichbarkeitsgraphen

3.1 Abstraktion

Anhand der Anzahl der Knoten und der Anzahl der Kanten dieses Erreichbarkeitsgraphen lässt sich leicht erkennen, dass dieser viel zu groß ist, als dass man an ihm spezielle Systemeigenschaften direkt ablesen könnte. Aus diesem Grund werden im Nachfolgenden eigenschaftserhaltende Abstraktionen (sogenannte schlichte Homomorphismen) auf den Erreichbarkeitsgraphen angewendet, um aus dem konkreten Systemverhalten (i. Allg. wesentlich einfachere) abstrakte Systemverhalten zu gewinnen.

Auf Basis der Aktionssprache können Abstraktionen des Verhaltens eines APA mittels Sprachhomomorphismen, genauer mittels alphabetischer Sprachhomomorphismen, $h : \Sigma^* \to \Sigma'^*$ formalisiert werden.

Durch solche Homomorphismen werden gewisse Zustandsübergänge ausgeblendet und andere umbenannt, was bedeuten kann, dass unterschiedliche Zustandsübergänge nicht mehr unterschieden werden.

Eine Abbildung $h : \Sigma^* \to \Sigma'^*$ heißt *Sprachhomomorphismus*, falls $h(\epsilon) = \epsilon$ und $h(yz) = h(y)h(z)$ für jedes $y, z \in \Sigma^*$. Er heißt *alphabetisch*, falls $h(\Sigma) \subset \Sigma' \cup \{\epsilon\}$.

Es stellt sich jetzt die Frage, in wie weit das abstrakte Systemverhalten das konkrete Verhalten widerspiegelt. Dabei kann das Problem auftreten, dass durch die Abstraktion fehlerhaftes Teilverhalten durch korrektes überdeckt wird. Geeignete Bedingungen an die Abstraktion verhindern dieses Problem [ORR00a].

Bekanntlich gibt es zwei Arten von Systemeigenschaften: *Sicherheitseigenschaften* (es geschieht nichts Falsches) und *Lebendigkeitseigenschaften* (irgendwann geschieht etwas Erwünschtes) [AS85].

Wegen der Lebendigkeitsaspekte müssen Systemeigenschaften durch ω-Sprachen (Mengen von unendlich langen Wörtern) formalisiert werden. Daher muß bezüglich der Erfülltheit von Eigenschaften "unendliches" Systemverhalten betrachtet werden.

Das geschieht durch sogenannte Eilenberg-Limites von Aktionssprachen (genauer: durch Eilenberg-Limites von modifizierten Aktionssprachen bei denen maximale Wörter durch unbeschränkte Wiederholung von Dummy-Aktionen fortgesetzt werden) [NO96].

Ohne zusätzlichen Fairness-Bedingungen ist das übliche Konzept des linearen Erfüllens von Eigenschaften (jedes unendliche Systemverhalten ist Element der Eigenschaft) nicht adäquat. Statt dessen formulieren wir eine abgeschwächte Erfüllbarkeitsrelation, welche die Möglichkeit von erwünschten Aktionen ausdrückt. Die Erfüllbarkeit solcher "Möglichkeits-Eigenschaften" wird im Begriff des *approximativen Erfüllens von Eigenschaften* definiert [NO96].

Ein System *erfüllt approximativ* eine Eigenschaft, wenn jedes endliche Systemverhalten zu einem unendlichen Systemverhalten fortgesetzt werden kann, welches die Eigenschaft linear erfüllt.

Für Sicherheitseigenschaften sind lineares und approximatives Erfüllen äquivalent [NO96].

Um approximativ erfüllte Eigenschaften eines Systems aus entsprechenden Eigenschaften seines abstrakten Verhaltens ableiten zu können, wird eine zusätzliche Eigenschaft der Abstraktion, die *Schlichtheit von Homomorphismen*, benötigt [Och94]. Dies ist eine sehr technische Bedingung bezüglich der möglichen Fortsetzungen endlicher Systemverhaltensweisen.

Für reguläre Sprachen ist die Schlichtheit von Homomorphismen entscheidbar. Basierend auf den Zusammenhangskomponenten der entsprechenden Automaten ist in [Och94] eine einfach zu überprüfende hinreichende Bedingung für die Schlichtheit angegeben.

Insbesondere ist auf einem streng zusammenhängenden Erreichbarkeitsgraph jeder Homomorphismus schlicht.

Der folgende Satz [NO96] zeigt, dass bezüglich der Verifikation von Systemeigenschaften approximatives Erfüllen und Schlichtheit von Homomorphismen genau zusammenpassen:

Schlichte Homomorphismen definieren genau die Klasse von Abstraktionen, für welche jede Eigenschaft vom abstrakten Systemverhalten genau dann approximativ erfüllt wird, wenn die "entsprechende" Eigenschaft vom konkreten Systemverhalten approximativ erfüllt wird.

Formal wird dabei die "entsprechende" Eigenschaft durch das inverse Bild der abstrakten Eigenschaft bzgl. des Homomorphismus ausgedrückt.

3.2 Sicherheits- und Lebendigkeitseigenschaften

Da der betrachtete Erreichbarkeitsgraph streng zusammenhängend ist (Nachweis mittels des SH Verification Tool), ist nach Aussage des letzten Abschnitts jeder auf ihm definierte Homomorphismus schlicht. Mit Hilfe einer solchen eigenschaftserhaltenden Abstraktion soll beispielsweise die folgende Frage beantwortet werden:

"Welche Aktionen können von welchem Subjekt auf dem Objekt *Aufenthaltsdokument_1* ausgeführt werden ?"

Zu diesem Zweck müssen alle Aktionen, die sich nicht auf dieses Objekt beziehen, durch die Abstraktion ausgeblendet werden. Im Beispielszenario verwenden wir hierzu ein Prädikat, welches die betreffenden Kanten im Erreichbarkeitsgraphen, die sich auf *Aufenthaltsdokument_1* beziehen spezifiziert und dann die Kanten, die nicht diesem Prädikat genügen, auf ϵ abbildet.

Nach Definition einer solchen Abstraktion berechnet das SH Verification Tool daraus eine minimale Repräsentation des abstrakten Systemverhaltens in Form eines sogenannten Minimalautomaten (siehe Abb. 4).

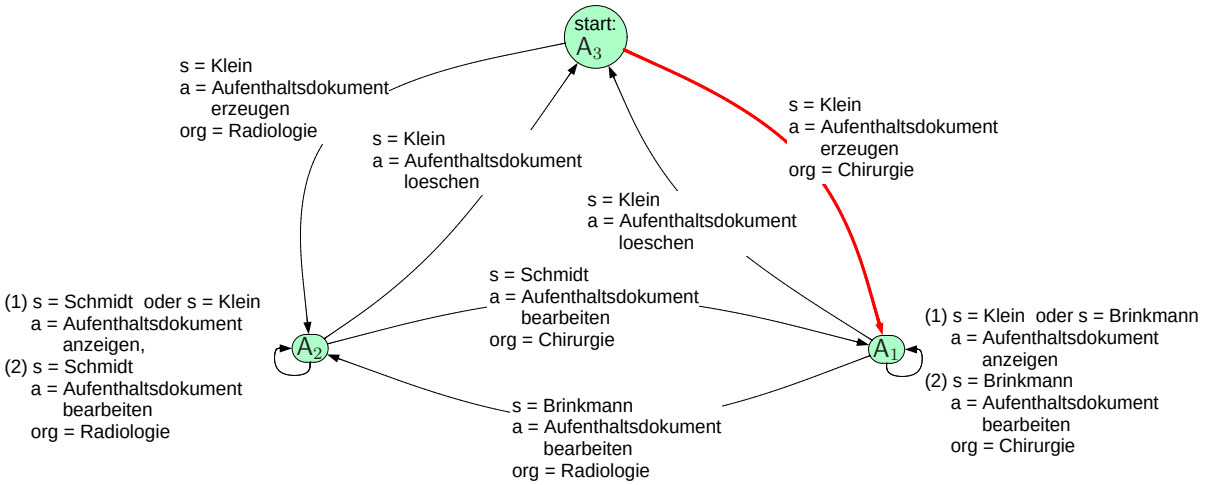

Abb. 4: Minimale Repräsentation des abstrakten Systemverhaltens

Der Automat in Abbildung 4 weist beispielsweise die folgenden Sicherheitseigenschaften nach:

- Das *Aufenthaltsdokument_1* kann nur durch den Verwaltungsmitarbeiter *Klein* erzeugt werden.
- Bei der Erzeugung wird entweder *Chirurgie* oder *Radiologie* in das Dokument eingetragen.
- Danach kann nur der Arzt der entsprechenden Fachabteilung dieses Dokument bearbeiten. Diese Eigenschaft wurde durch die folgenden Regeln im Or-BAC Modell realisiert:

  Permission(Chirurgie, Arzt, Aufenthalt_aendern, Aufenthalt,
  Aufenthaltsdokument_belegt, Ort_stimmt_ueberein)
  Permission(Radiologie, Arzt, Aufenthalt_aendern, Aufenthalt,
  Aufenthaltsdokument_belegt, Ort_stimmt_ueberein)

- Wenn sich der im Dokument eingetragene Aufenthaltsort durch die Bearbeitung ändert, wechselt entsprechend der Arzt, der das *Aufenthaltsdokument_1* anzeigen und bearbeiten kann.
- Der Verwaltungsmitarbeiter *Klein* ist der Einzige, der dieses Dokument auch löschen kann.

Darüber hinaus sieht man auch folgende Lebendigkeitseigenschaft:

- Herr *Klein* kann immer wieder einmal ein *Aufenthaltsdokument_1* mit der Organisationseinheit *Chirurgie* erzeugen.

Man erkennt diese Eigenschaft im Automaten der Abbildung 4 durch die Kante (3)

$$(A_3, (s = Klein, a = Aufenthaltsdokument\ erzeugen, org = Chirurgie), A_1) \quad (3)$$

die immer wieder erreicht werden kann. Da die Abstraktion zur Erzeugung dieses Automaten ein schlichter Homomorphismus ist, was mittels des SH Verification Tool nachgewiesen wurde, ist die "entsprechende" Lebendigkeitseigenschaft auch im Erreichbarkeitsgraphen gegeben. Diese Eigenschaft ist dort über 944 unterschiedliche Kanten repräsentiert. Eine davon ist die Kante (2) in Abbildung 3.

Außer diesen Eigenschaften lassen sich auch alle anderen Eigenschaften des Minimalautomaten in entsprechende Eigenschaften des Erreichbarkeitsgraphen übersetzen.

4 Ausblick

Wir möchten an dieser Stelle zunächst Herrn Christoph Breker danken, der in seiner Diplomarbeit [Bre07] das hier vorgestellte Beispiel ausführlich untersucht hat und weitere Eigenschaften dazu nachgewiesen hat. Es wird dort beispielsweise auch aufgezeigt, wie man Fehler in der Spezifikation der Zugriffskontrollpolitik anhand von Minimalautomaten finden kann.

Im Kontext des hier gewählten Szenarios sind die nachzuweisenden Eigenschaften Sicherheits- (in Sinne von "safety") und Lebendigkeitseigenschaften. Komplexere Sicherheitseigenschaften wie Authentizität und Vertraulichkeit können mit den hier verwendeten Methoden ebenfalls betrachtet werden [GOR04].

Da traditionelle Modell-Checking Techniken die Verifikation eines gewünschten Sytemverhaltens wegen der kombinatorischen Zustandsexplosion nur für Systeme mit wenigen Komponenten ermöglichen, haben wir weiterhin eine abstraktionsbasierte Methode entwickelt, die es für Systeme mit vielen identischen Komponenten erlaubt, deren Verhalten unabhängig von der Anzahl der Komponenten zu verifizieren [OR07].

Literatur

[ABB+03] A. Abou El Kalam, R. El Baida, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, A. Miège, C. Saurel, and G. Trouessin. Organization Based Access Control. In *4th IEEE International Workshop on Policies for Distributed Systems and Networks (Policy'03)*, June 2003.

[AS85] B. Alpern and F. B. Schneider. Defining liveness. *Information Processing Letters*, 21(4):181–185, October 1985.

[Bre07] Christoph Breker. Formale Modellierung und Analyse einer OrBAC-basierten Sicherheitspolitik am Beispiel der elektronischen Krankenakte. Diplomarbeit, Universität Frankfurt, 2007.

[Cau08] Dr. Jörg Caumanns. Übergreifendes Sicherheitskonzept für Umsetzung und Betrieb elektronischer Fallakten. Spezifikation, eFA Konsortium, 2008. © eFA Konsortium (http://www.fallakte.de).

[fdD] Der Bayerische Landesbeauftragte für den Datenschutz. Orientierungshilfe: Technisch-organisatorische Forderungen an ein benutzer- und datenschutzfreundliches Patientenverwaltungssystem bzw. Krankenhausinformationssystem (KIS), Stand: 26.01.2005. Technical report.

[GOR04] S. Gürgens, P. Ochsenschläger, and C. Rudolph. On a formal framework for security properties. *International Computer Standards & Interface Journal (CSI), Special issue on formal methods, techniques and tools for secure and reliable applications*, 2004.

[NO96] U. Nitsche and P. Ochsenschläger. Approximately satisfied properties of systems and simple language homomorphisms. *Information Processing Letters*, 60:201–206, 1996.

[Och94] P. Ochsenschläger. Verification of cooperating systems by simple homomorphisms using the product net machine. In J. Desel, A. Oberweis, and W. Reisig, editors, *Workshop: Algorithmen und Werkzeuge für Petrinetze*, pages 48–53. Humboldt Universität Berlin, 1994.

[OR07] P. Ochsenschläger and R. Rieke. Abstraction Based Verification of a Parameterised Policy Controlled System. In *Computer Network Security, Fourth International Conference on Mathematical Methods, Models and Architectures for Computer Network Security, MMM-ACNS 2007, St. Petersburg*, volume 1 of *CCIS*. Springer, September 2007. © Springer.

[ORR00a] P. Ochsenschläger, J. Repp, and R. Rieke. Abstraction and composition – a verification method for co-operating systems. *Journal of Experimental and Theoretical Artificial Intelligence*, 12:447–459, June 2000. Copyright: ©2000, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

[ORR00b] P. Ochsenschläger, J. Repp, and R. Rieke. The SH-Verification Tool. In *Proc. 13th International FLorida Artificial Intelligence Research Society Conference (FLAIRS-2000)*, pages 18–22, Orlando, FL, USA, May 2000. AAAI Press.