Implementierung von Security Policies in offenen Telekollaborationen

Michael Herfert · Andreas U. Schmidt[1] ·
Peter Ochsenschläger · Jürgen Repp · Roland Rieke[2] ·
Martin Schmucker[3] · Steven Vettermann[4] · Uwe Böttge[5] ·
Cristina Escaleira · Dirk Rüdiger[6]

[1]Fraunhofer Institut Sichere Telekooperation, Dolivostraße 15, 64293 Darmstadt
{herfert, aschmidt}@sit.fraunhofer.de

[2]Fraunhofer Institut Sichere Telekooperation, Rheinstraße 75, 64295 Darmstadt
{ochsenschlaeger, repp, rieke}@sit.fraunhofer.de

[3]Fraunhofer Institut Graphische Datenverarbeitung, Fraunhoferstraße 5, 64283 Darmstadt
mschmuck@igd.fraunhofer.de

[4]Fachbereich Maschinenbau, DiK, TU Darmstadt, Petersenstr. 30, 64287 Darmstadt
vetterm@dik.maschinenbau.tu-darmstadt.de

[5]Fraunhofer Institut Produktionsanlagen&Konstruktionstechnik, Pascalstr. 8-9, 10587 Berlin
Uwe.Boettge@ipk.fraunhofer.de

[6]Zentrum für Graphische Datenverarbeitung, Joachim-Jungius-Strasse 11, 18059 Rostock
{came, drue}@rostock.zgdv.de

Zusammenfassung

Die besonderen Anforderungen der Planung und Durchsetzung von Sicherheitspolitiken in Telekollaborations-Umgebungen, in die verschiedene Partner unterschiedliche Sicherheitsrichtlinien einbringen, werden diskutiert. Ein durchgängiges Vorgehensmodell wird vorgestellt, dass von der Planung und Formalisierung über die formale Verifikation bis zur Implementierung der Sicherheits-Policies reicht. An einem konkreten Szenario aus der Automotive Industrie wird exemplarisch die Umsetzung des Konzepts dargestellt. Schließlich werden verbleibende Probleme diskutiert und bewertet.

1 Motivation

Die Implementierung von Security Policies in offenen Telekooperationsumgebungen wie etwa Telekonferenzen geht über die primären Ziele einfacher Zugangskontrolle hinaus. Die Aufgabe verlangt nach einem flexiblen Rahmenwerk, einem **Policy Implementation**

Framework (PIF), das folgende voneinander abhängige Aufgaben zu erfüllen hat:

1. **Definition**: Ausgehend von informellen Beschreibungen von Sicherheitsrichtlinien müssen Policies in strikte, logische Form gebracht und in einem interoperablen Datenformat dargestellt werden.
2. **Verifikation**: An einem formalen Zustandsmodell des betrachteten Systems ist die Einhaltung der Policies zu überprüfen, um Sicherheitsbeweise zu erhalten.
3. **Enforcement**: Es sind Komponenten zu entwickeln und an das System anzubinden, die gegebenenfalls die Einhaltung der Policies erzwingen.

Bei der Realisierung des PIF sollen zudem wichtige Randbedingungen und praktische Ziele berücksichtigt werden:

4. **Offenheit**: Parteien können zur Telekollaboration hinzutreten und sie verlassen und bringen dabei jeweils eigene, heterogene Sicherheitsanforderungen ein.
5. **Anwendungsunabhängigkeit**: Das PIF soll auf eine möglichst große Klasse von existierenden und zukünftigen Telekollaborationsanwendungen anwendbar sein, die eventuell nicht über eigene Sicherheitsmechanismen verfügen.
6. **Ergonomie**: Das Enforcement von Policies ist Nutzern durch geeignete Awareness-Komponenten transparent zu machen, um Sicherheitsbewusstsein zu fördern.

In ihrer Gesamtheit stellen die angeführten Punkte eine Herausforderung dar, die nur mit Hilfe grundlegend neuer Methoden angegangen werden kann, was die Autoren dieses Artikels im Projekt MakoSi [1] unternommen haben. In der Literatur werden vergleichbare Projekte, wie zum Beispiel [2] üblicherweise mit den bekannten Methoden des Role Based Access Control (RBAC) [3] oder Verallgemeinerungen davon [4] behandelt. Die besonderen Anforderungen an Offenheit und starke formale Verifikation schließen dies in unserem Fall jedoch aus, insbesondere da RBAC oder zum Beispiel auch Ponder Policy Languages nicht ohne weiteres ein Systemmodell liefern. Zudem ist RBAC besser geeignet für die Beschreibung hierarchisch strukturierter Policy-Mengen in Szenarien mit einer wohldefinierten Zuordnung von Berechtigungen zu Rollen, nicht aber für die heterogenen Mengen von Policies, die wir betrachten. Auch das in [5] vorgestellte Konzept der Metapolitiken für die domänenübergreifende Zugriffssteuerung liefert unter den gegebenen Randbedingungen keinen praktikablen Ansatz zu Lösung des Problems, obwohl sich seine Ziele und grundlegende Methoden weitgehend mit unseren decken. Die dort aufgezeigte Zugriffssteuerung basiert auf in folgender Form formulierten Politiken: "ein Subjekt führt eine Operation auf einem Objekt aus" und der Fragestellung "ist diese Operation zulässig". Im vorliegenden Fall ist jedoch allgemeiner zu klären: "Ist ein Zustandswechsel innerhalb des betrachteten Systems unter den gegebenen Randbedingungen zulässig?". Einen wesentlichen Schwerpunkt bildet hierbei auch die zügige und praktikable Integration sowohl der Lösung als auch variabler Sicherheitspolitiken in bestehende Standard-IT-Architekturen.

Spezielle Schwierigkeiten entstehen aus dem inhärenten Konflikt zwischen den Zielen 4 und 5 und der Forderung 2 nach Sicherheitsbeweisen: Das Hinzufügen von Policies zum System zur Laufzeit kann bereits erzielte Beweise wieder in Frage stellen, ebenso wie es eine dynamische Veränderung des zugrundeliegenden Telekollaborationssystems tun würde. Aus diesen Gründen grenzen wir unsere Thematik weiter ein und behandeln einen spe-

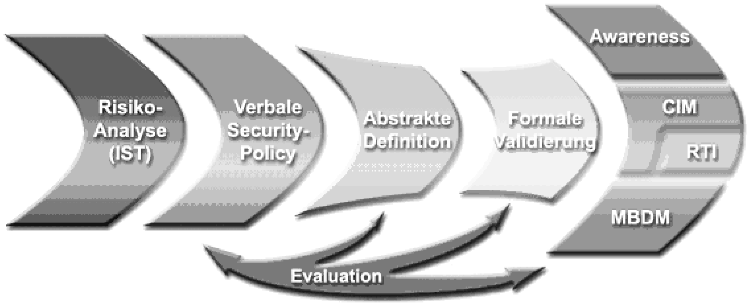

Abb. 1: Meilensteine des Vorgehensmodells von MakoSi.

ziellen Anwendungsfall: Die Implementation von Security Policies in einem graphischen Telekonfernz-Tool für Konstruktionsanwendungen in einem aus der Automobilindustrie abgeleiteten Szenario. Konkreter use-case ist hier das iViP-Whiteboard Telekonferenz-Tool [6, 7]. In dieser Einschränkung ist unser Beitrag nur als proof-of-concept zu verstehen, der jedoch weitergehende Anwendungsmöglichkeiten offenläßt.

2 Vorgehensmodell und Methodik

Ziel des Vorgehensmodells ist es, domänenübergreifende Zusammenarbeit zu realisieren und alle Beteiligten von der Spezifikation bis hin zur Implementierung der in diesem Kontext geltenden Security-Policies methodenbasiert zu unterstützen. Dies trägt dazu bei, dass etwaige Realisierungen schneller, qualitativ hochwertiger und für Dritte nachvollziehbar gestaltet werden können. Zu diesem Zweck sind die in der Abbildung 1 beschriebenen Meilensteine zu durchlaufen. Das Vorgehensmodell basiert auf modernen System- und Software-Engineering-Methoden und beschreibt ein iteratives und inkrementelles Vorgehen. Evaluationen dienen der Qualitätssicherung in jedem Schritt.

Basierend auf existierenden Standards, etwa [8], Guidelines, Kategorisierungen etc., werden die Anwender zuerst bei der Durchführung einer Risiko-Analyse und, darauf aufbauend, bei der Spezifikation von verbal beschriebenen Security-Policies im Kontext von domänenübergreifender Zusammenarbeit unterstützt. Diesem Schritt folgt eine ausführliche Beschreibung, wie die Anwender die so erarbeiteten Security-Policies abstrahieren und schließlich in eine formales Modell überführen können. Dieses formale Modell dient als Grundlage von Sicherheitsbeweisen. Neben der Überprüfung der IT-Sicherheit, dient die formale Validierung auch als zusätzliche Qualitätssicherungsmaßnahme. Aufbauend auf den so erlangten Resultaten, erfolgt im letzten Schritt die Realisierung von Anwendungen zur Security-Policy-konformen, domänenübergreifenden Zusammenarbeit. Im Folgenden werden die Merkmale des entstehenden Policy-Driven-Systems detailliert beschrieben, wobei auch näher auf die Komponenten des letzten Meilensteins, der den konkreten Implementierungsteil des PIF darstellt, eingegangen wird. Die Methode zur Anwendung des Vorgehensmodells in einem konkreten Fall besteht aus den Schritten

1. **Analyse**: Aus semi-formalen Beschreibungen wird ein erstes formales Systemmodell gebildet und auf die Einhaltung von Beispiel-Policies hin überprüft.
2. **Definition des Datenmodells**: Als Grundlage des weiteren Vorgehens wird ein

Basis-Datenmodell gebildet. Daraus wird ein unabhängiges Systemmodell abgeleitet, das das Policy-Enforcement berücksichtigt. Seine Analyse liefert die gewünschten Sicherheitsbeweise.

3. **Implementation**: Schließlich werden Enforcement-Komponenten generiert, mit Awareness-Komponenten verknüpft und an das System angebunden.

Dabei bauen die drei Schritte in einem iterativen Prozeß aufeinander auf. Die Analyse eines komplexen Szenarios wird häufig mit einem semi-formalen Modell, zum Beispiel in Form von UML-Diagrammen starten. Ein solches Modell wird fast immer Lücken und Inkonsistenzen aufweisen, die am abgeleiteten formalen Zustandsmodell in Schritt 1 aufgedeckt werden können. Ist das Modell entsprechend komplettiert und bereinigt, läßt sich an ihm sodann die *prinzipielle Implementierbarkeit* der Sicherheitsrichtlinien nachweisen. Die in Schritt 1 gewonnene Information dient in Schritt 2 wesentlich zur Schaffung des MakoSi Basis-Datenmodells (MBDM), das das informationale Rückgrat des MakoSi PIF und die Grundlage zur (semi-) Automatisierung der folgenden Schritte bildet. Insbesondere wird ein rollenbasiertes Systemmodell entwickelt und in einem interoperablen XML-Format abgelegt. Ein systemspezifisches XML-Format zur Formulierung der Policies baut auf dem MBDM auf. Aus diesen beiden Datensätzen wird das zweite formale Systemmodell generiert, an dem dann die formalen Sicherheitsbeweise durchgeführt werden die zeigen, dass die zu realisiernden Enforcement-Komponenten die Policies *tatsächlich* implementieren. Diese Komponenten werden in Schritt 3 — ebenfalls auf Grundlage des MBDM — konstruiert und an das System angebunden.

3 Das Whiteboard Szenario

Das iViP Whiteboard (WB) ist ein Telekonferenz-Tool das eine virtuelle Zeichenfläche für Konstruktionsaufgaben bereitstellt. Das Whiteboard als solches bietet nur rudimentäre Sicherheitsmechanismen und stellt damit ein geeignetes Beispiel für Policy-Implementierung dar. Das WB basiert auf einem Client-Server Modell in dem alle Benutzer die gleiche Client-Software benutzen. Obwohl es wünschenswert wäre, können wir hier aus Platzgründen nicht näher auf das Whiteboard eingehen und verweisen auf [6, 7].

In unserem Szenario wollen mehrere Benutzergruppen, genannt **Rollen**, ihre unterschiedlichen Security-Policies im System befolgt wissen. Eine asymmetrische Rollenverteilung besteht nun etwa zwischen einem **Automobilhersteller** (A) und einem **Zulieferer** (Z), letztlich resultierend aus ökonomischen Machtverhältnissen, die dazu führen, dass (Z) mehr Daten in das WB einbringen muß und (A) mehr Daten geheimzuhalten wünscht. Die dritte wesentliche Rolle bildet der **Whiteboard Server** (W) selbst. Während des Ablaufs einer WB-Sitzung führen die **Rollenspieler**, das heißt die konkreten Instanzen einer Rolle verschieden Aktionen durch, zum Beispiel das Initiieren einer neuen Konferenz, das Beitreten zu einer bestehenden Konferenz oder ihr Verlassen, oder das Senden eines Dokuments. Diese Aktionen sind letztlich die Punkte, an denen die Einhaltung der Policies überprüft, beziehungsweise erzwungen werden muss.

Wir nehmen für unser Beispiel folgende, recht überschaubare Menge von Policies an, die jeweils eindeutig einer der Rollen zugeordnet werden können. Die informellen Policies von A und Z sind:

PA1 ‘Originaldaten dürfen nicht herausgegeben werden’ und

PA2 ‘Daten dürfen nur mit bekannten Partnern ausgetauscht werden’.

PZ1 ‘Daten dürfen nur an bekannte Partner gesendet werden’,

PZ2 ‘Transfer von Originaldaten muss verschlüsselt erfolgen’ und

PZ3 ‘jeder Datentransfer ist zu protokollieren’.

Die Unterscheidung der Daten in “original” und “nicht original” ist hier zunächst künstlich und kann für eine beliebige Klassifizierung der Daten stehen.

4 Zustandsmodell und erste Verifikation

Für die gesamte formale Modellierung des Systems verwenden wir mit den sogenannten **Asynchronen Produktautomaten** (APA) [9, 10] eine flexible operationale Beschreibung kooperierender Systeme, die der Theorie der formalen Sprachen entspringt. Die Analyse von APAs wird unterstützt durch das SH-verification Tool (SHVT [11]) das Komponenten für den kompletten Zyklus von formaler Spezifikation bis hin zu umfassender Validierung besitzt. Das SHVT ist zum Beispiel in [11] ausführlicher beschrieben.

Wie jeder formale Sicherheitsbeweis kann aber auch die Verifikation von Policies im Rahmen von APAs nur unter bestimmten Annahmen stattfinden, die den Gültigkeitsbereich der Beweise abstecken. Im WB-Szenario sind dies Annahmen hinsichtlich der konkreten Implementierung der Policies, zum Beispiel dass Policies bei jedem Teilnehmer vollständig erzwungen werden und dass der Server W die nötigen Daten für die Erzwingung der Policies zu jedem Zeitpunkt vorhält. Zudem wird die Authentisierung von Teilnehmern sowie das Management von Teilnehmergruppen nicht berücksichtigt.

Unter Verwendung der grafischen Eingabekomponenten des SHVT wird ein vereinfachtes Zustandsmodell aller drei WB-Rollen aus einer semi-formalen Beschreibung seiner Funktionalität durch UML-Diagramme abgeleitet. Das volle WB APA Modell kann zerlegt werden in die Teile, die zu A, Z und W gehören und gemäß der Aktionsklassen, zum Beispiel der Basisfunktionen `create`, `join`, `leave`, `ssl`, `read/write` und so weiter.

Die Simulation der vollen Systemdynamik erfordert ca. eine Million elementarer Übergänge, und enthüllte einige Modellierungsfehler (deadlocks) in frühen Versionen. Um das Systemmodell mit dem semi-formalen UML-Modell abzugleichen, wurden Abstraktionen mit Hilfe des Morphismus-Editors des SHVT definiert und der zugehörige Minimalautomat berechnet. Für das lokale Verhalten des WB-Clients ergaben sich Minimalautomaten (<50 Zustände), die in allen wesentlichen Punkten mit dem UML-Modell übereinstimmen. Die informellen Policies PA1–PA2, PZ1–PZ3 müssen nun für den ersten Verfikationsschritt “von Hand” in das APA-Modell eingebracht werden, in Form von temporalen Logikformeln. Ihre Abarbeitung im APA-Modell zeigt einerseits mögliche Sicherheitslücken im System, andererseits identifiziert man so die Punkte, an denen *Policy-relevante* Aktionen ausgeführt werden, also die Punkte, an denen später die Enforcement-Komponenten angreifen.

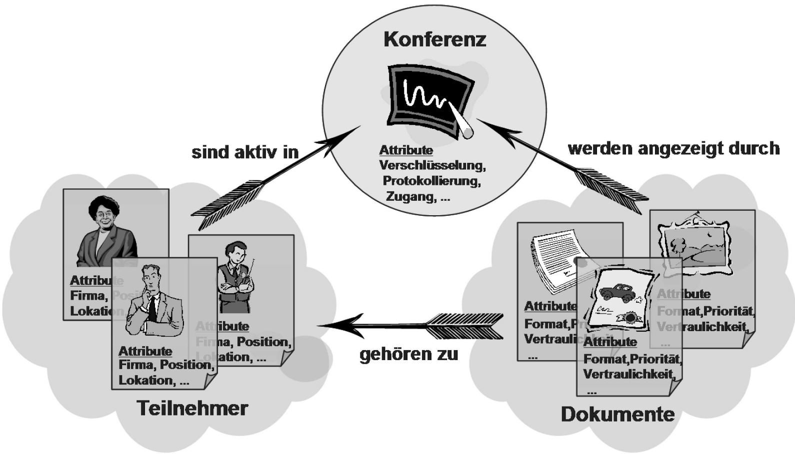

Abb. 2: Basis-Semantik einer Kollaboration.

5 Basis-Datenmodell

Fundament des MakoSi PIF ist eine durchspezifizierte Datenstruktur, die auf einer einheitlichen Semantik basiert. Diese Struktur muß sicherstellen, dass die in der formalen Evaluierung, in den Enforcement- und in den Awareness-Komponenten verwendeten Modelle des Systems nicht auseinanderklaffen. Sie wird durch das MakoSi-Basis-Datenmodell (MBDM) geschaffen. Es basiert auf einem einfachen, rollenbasierten Modell offener Telekooperationsumgebungen, das nur die drei Objekte Konferenz (**Con**), Teilnehmer (**Mem**) und Dokument (**Doc**) enthält. Diese drei grundlegenden Rollen, genannt Rollen-Typen, werden jeweils durch ihre Eigenschaften, sogenannte Attribute (**Att**) charakterisiert und stellen paradigmatisch die möglichen Verhaltensweisen von Objekten in einer Telekollaborationsumgebung dar. Die Rollentypen enthalten dabei nur wenige Attribute, die einige grundlegende semantische Beziehungen zwischen ihnen ausdrücken, insbesondere die eindeutige Zuordnung von Dokumenten zu Teilnehmern (Besitz) und das Vorhandensein eines Teilnehmers beziehungsweise eines Dokuments in einer Konferenz. Die wesentliche und für die Security-Policies relevante Annahme über die Dynamik des Systems ist nun, dass ein Dokument, das in eine Konferenz eingebracht wird, allen anderen Teilnehmern in dieser Konferenz unmittelbar zur Verfügung steht (Abb. 2).

Die drei Rollentypen reichen zur Modellierung eines komplexen Systems nicht aus. Das Modell kann daher um Rollen erweitert werden, die die Grundeigenschaften eines Rollentyps (seine Attribute) erben und weitere hinzufügen — etwa Klassifikation eines Dokuments, Verschlüsselungsstatus und ähnliches. In der detaillierten Definition aller Rollen und ihrer Attribute steckt nun die gesamte in MakoSi verwendete Information über den Aufbau des Systems. Auf der untersten, das heißt der Implementationsebene der Enforcement-Komponenten von MakoSi, wird das Rollenmodell durch Instantiierung der einzelnen Rollen — durch Rollenspieler — zur Laufzeit konkretisiert.

Die Dynamik des Modells besteht nun in der Änderung von Werten der einzelnen Attribute der Rollen während des Ablaufs einer Kollaboration. Dies ist auch die Grundlage für die Formulierung von Policies, die mit Hilfe der Attribute als Event-Condition-Action (ECA) Regeln dargestellt werden. Das heißt, ein Ereignis (Event) im System, entsprechend einer bestimmten Attributsänderung, stößt die Auswertung logischer Bedingungen (Conditions)

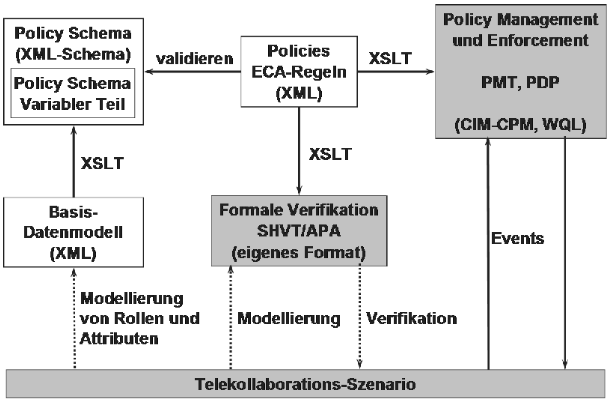

Abb. 3: Basis-Datenmodell und Informationsfluss in MakoSi.

auf dem gegenwärtigen Zustand aller Attribute an, und dies löst — im Fall positiven Ergebnisses — gewisse Aktionen (Actions) aus, zum Beispiel das Senden eines Dokuments an das Whiteboard bei gleichzeitigem Aufruf einer Awareness-Komponente.

Konkret ist das MBDM als XML-Datenstruktur realisiert. In ihr werden die Namenskonventionen für Rollen und Attribute sowie deren Typisierung mittels Standarddatentypen festgelegt. Damit ist das MBDM das Rückgrat der informationellen Infrastruktur von MakoSi. Im weiteren Ablauf des MakoSi-Prozesses wird aus dem MBDM ein XML-Schema generiert, das die genaue Struktur aller (im gegebenen System) formulierbaren Policies festlegt. Die Policies können nun verwendet werden einerseits zur semi-automatischen Generierung von Enforcement-Komponenten, wie auch andererseits zur Bildung des formalen Systemmodells. Die dazu nötigen Schnittstellen zwischen den verschiedenen Datenformaten werden durch XSL Transformations realisiert (Abb. 3).

Eine zweite, unabhängige Verifikation wird nun an dem aus dem MBDM generierten APA-Modell durchgeführt. Dabei werden die Enforcement-Komponenten ohne Berücksichtigung ihrer internen Struktur als "black boxes" eingebunden. Dies liefert schließlich die geforderrten Sicherheitsbeweise.

6 Implementierung mittels CIM/WBEM

Grundlage für die Implementierung der Enforcement-Komponenten bildet das Common Information Model (CIM) der DMTF [12] für die Darstellung von Management-Informationen in verteilten Softwareumgebungen. CIM spezifiziert ein Standardmodell für die neutrale Beschreibung einer Vielzahl von IT-Komponenten zum Zweck ihrer Administration. Die Web Based Enterprise Management (WBEM) Initiative verwendet CIM zur Entwicklung einer Software-Architektur, die ein plattformunabhängiges Management in heterogenen IT-Landschaften ermöglichen soll. Firmen wie Microsoft oder Sun Microsystems arbeiten in dieser Initiative mit und nutzen diese Technologie innerhalb ihrer (Betriebs)systeme. Dieser Hintergrund und die Tatsache, das CIM auch Objektstrukturen zur Abbildung von Policy-Regeln enthält, machen eine Verwendung von CIM/WBEM im Rahmen von MakoSi interessant. Ausgehend von der Annahme, dass ein beliebiger realer Systemzustand über diese WBEM Services Architektur abgefragt bzw. beeinflusst werden kann, stellt diese Technologie eine geeignete Grundlage für das zur realisierende

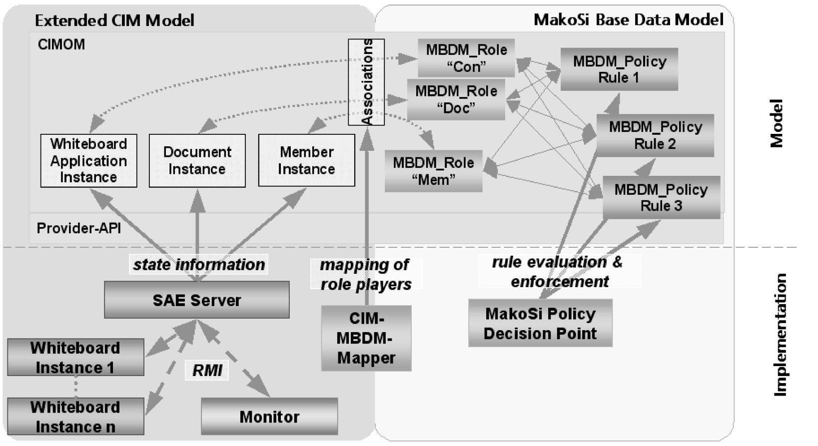

Abb. 4: Verknüpfung zwischen MBDM-Rollen und den realen Rollenspielern

Enforcement und die Awareness-Vermittlung dar.

Für die Implementation der Enforcement-Komponenten von MakoSi wählten wir das CIM Policy Model [13] (CCPM), das auf dem IETF Policy Core Information Model [14] basiert. Ein Schema für die Darstellung von CIM Daten in XML ist in [15] definiert. Obwohl es für das Policy-Enforcement durch existierende Software-Tools gut geeignet ist, kann das CCPM nicht direkt zur Formulierung der ECA-Policies verwendet werden (zum Beispiel kennt es nicht das Event-Konzept), was ein weiterer Grund für die Darstellung der Policies in einem eigenen XML-Format war. Die aus den MBDM-XML-Policies generierten CIM-Polcies werden in einem standard container, dem **CIM-Repository**, abgespeichert.

Der erste Schritt ist nun die Abbildung des MBDM in eine geeignete Repräsentationsform im CIM. Die Abbildung erfolgt in zwei Teilbereichen. Den einen Bereich bildet ein konkretes Rollenmodell (Szenario), in dem die Rollen und ihrer Attribute beschrieben sind und der andere Bereich umfasst die für dieses Szenario modellierten Policy-Regeln. Zur Abbildung der Rollen wurden eigene Objektklassen im CIM-Schema definiert, während für die Regeln die vorhandenen Policy-Objekte nur minimal erweitert werden mussten. Nach dem Import des Szenarios (MBDM-Rollen) liegen die Rollen als instantiierbare Klassen im CIM vor und die Regeln (MBDM-Regeln) sind in Form von Policy-Instanzen repräsentiert. Damit kann die Konfiguration der policygesteuerten Laufzeitumgebung über die Zuordnung von Rollenspielern zu den MBDM-Rollen erfolgen. Diese Zuordnung wird im CIM über Assoziationsklassen vorgenommen, die CIM-Klassen, welche die Rollenspieler repräsentieren, mit den MBDM-Rollenklassen verbinden. Dadurch wird auch definiert, welches Attribut einer MBDM-Rolle durch welches Attribut eines Rollenspielers repräsentiert wird. Auf diese Art wird beispielsweise die Whiteboard-Applikation mit der MBDM-Rolle Konferenz verknüpft. Abbildung 4 zeigt schematisch diese Vorgehensweise. Im rechten Teil der Abbildung sind die CIM-Objekte dargestellt, die das MBDM-Szenario und die Policy-Regeln im CIM-Repository repräsentieren. Links ist angedeutet, wie das Whiteboard an die WBEM-Architektur angebunden ist.

Basierend auf Suns Services für WBEM [12], entwickelten wir ein graphisches Policy Management Tool (PMT), mit dem die CCPM Policies bearbeitet werden können, zum Beispiel um Aspekte der Policies auf der Ebene der Rollenspieler zu defnieren (die ja in

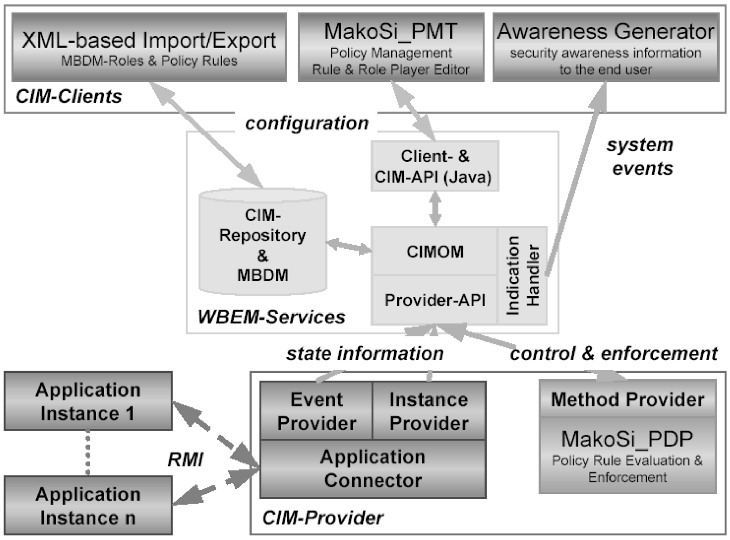

Abb. 5: Architektur für das Policy Enforcement.

der formalen Verifikation und dem MBDM vernachlässigt wurde), oder Informationen für Awareness-Komponenten einzugeben.

Neben diesem Administrationstool, das auch den Import der der MBDM-Rollen und Regeln sowie die Zuordnung zwischen Rollen und Rollenspieler unterstützen, gibt es für die Laufzeitumgebung zwei wesentliche Module, die auf die CIM/WBEM-Architektur aufsetzen:

- Der **CIM-MBDM-Mapper**, welcher in erster Linie für das Mapping zwischen dem MBDM und den Rollenspielern zuständig ist und
- Der **MakoSi Policy Decision Point** (PDP), welcher die Auswertung der Policy-Regeln übernimmt, die im Kontext des MBDM formuliert wurden.

Auf die Funktionsweise der beiden Module gehen wir im Folgenden ein. Die Architektur, in die sie eingebunden sind, zeigt Abbildung 5.

CIM-MBDM-Mapper

Der CIM-MBDM-Mapper ist dafür zuständig, dass der reale Systemzustand, repräsentiert durch Rollenspieler-Objekt im CIM, auf einen äquivalenten Zustand innerhalb der MBDM-Rollen abgebildet wird. D.h. für jeden Rollenspieler, der über eine Assoziation mit einer MBDM-Rolle verknüpft ist, muss gewährleistet werden, dass der Zustand der Rollenspielerinstanz synchron in einer MBDM-Rolleninstanz widergespiegelt wird, da nur so der PDP auf dem aktuellen Systemzustand Entscheidungen fällen kann. Um hier Konsistenz zu gewährleisten ist der CIM-MBDM-Mapper für die Instantiierung, Modifikation und Entfernung aller Instanzen aller MBDM-Rollenklassen zuständig. Alle hierfür notwendigen Informationen sind in dem CIM-Repository vorhanden. Ändert sich der Zustand einer Rollenspielerinstanz, so wird dies von dem Mapper registriert und der Zustand der assoziierten MBDM-Rolle nachgeführt. Darüber hinaus besitzt der Mapper eine einfache Schnittstelle für das Policy-Enforcement. Im Sinne eines Security-Managers kann über so genannte CIM-Methodenaufrufe eine Anfrage gestellt werden, ob ein bestimmter Zustandsübergang (Änderung eines Attributs) unter den gegebenen Randbedingungen

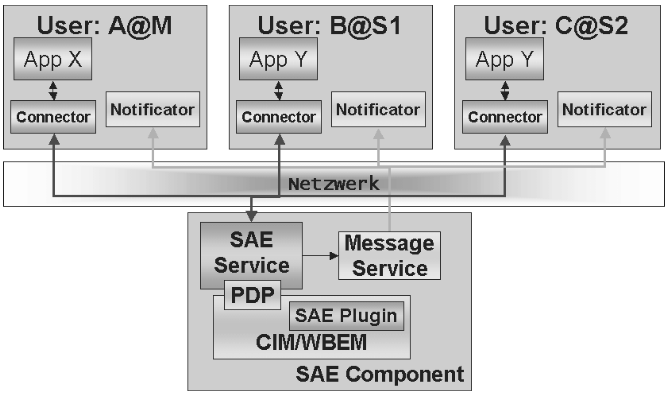

Abb. 6: Architektur der SAE-Komponente

zulässig ist oder nicht. Der Mapper übersetzt (mapped) die Anfrage auf den entsprechenden Zustandsübergang in den MBDM-Rollen und leitet diese dann an den PDP weiter, der die eigentliche Regelauswertung durchführt. Das Ergebnis der Regelauswertung wird durch den Mapper an die anfragenende Instanz zurückgegben.

MakoSi Policy Decision Point (PDP)

Der MakoSi Policy Decision Point ist für die Auswertung der Policies, d.h. der MBDM-Regeln zuständig. Für das Enforcement geschieht die Auswertung der Regeln auf Anfrage (pro-aktiv), d.h. über so genannte checkPolicy-Methoden (checkCreate, checkModify, und checkDelete). Der PDP wertet die Regelsätze aus liefert ein entsprechendes Ergebnis zurück. Die Ergebnismeldung enthält neben der Entscheidung, ob der Zustandsübergang zulässig ist oder nicht, auch Verweise auf die ausgewerteten Regeln. Bei einer negativen Entscheidung wird darüber hinaus eine Liste der Bedingungen mitgegeben, die nicht erfüllt wurden und somit zur Ablehnung führten. Dies dient als Ausgangspunkt für die Bereitstellung von Awareness-Informationen, um Policy-Entscheidungen gegenüber dem Anwender transparenter machen zu können. Der PDP wertet die Regeln jedoch nicht nur auf Anfrage aus, sondern ebenso wie der Mapper überwacht der PDP die Zustände der instantiierten MBDM-Rollen. So werden auch nicht autorisierte Zustandsübergänge festgestellt und Verstöße gegen die Policy können erkannt und gemeldet werden (reaktiv).

Die wesentlichen Vorteile des Lösungsansatzes auf Basis von CIM/WBEM sind:

- Wiederverwendung von validierten Policy-Regeln für verschiedene Rollenspieler ist ohne Aufwand möglich,
- Das Implementieren von Policy-Regeln beschränkt sich auf die Zuordnung von Rollenspielern zu MBDM-Rollen,
- Die Auswertung der Regeln durch den PDP erfolgt unabhängig von der jeweiligen Applikation (Rollenspieler) innerhalb des Kontextes der MBDM-Rollen,
- Nutzung von Technologien und Standards, die eine breite Anwendung in heterogenen IT-Landschaften ermöglicht und unterstützt.

7 Risikovermittlung beim Benutzer

Da Sicherheitslücken oft das Resultat der fehlerhaften Anwendung von entsprechenden Sicherheitsmaßnahmen durch die Benutzer sind, genügt die automatische Durchsetzung von Sicherheitsrichtlinien in realistischen Anwendungsszenarien nicht für die Gewährleistung eines hohen Sicherheitsstandards im Unternehmen. Der Mangel an Risikobewusstsein der Anwender im alltäglichen Umgang mit organisatorischen Sicherheitsregeln trägt zu den Sicherheitsverletzungen bei und bedroht Vertraulichkeit, Integrität und Gültigkeit von sensiblen Daten. Eine Sicherheitsrichtlinie kann nur effektiv sein, wenn die Anwender die notwendigen Sicherheitsmaßnahmen kennen, verstehen und anerkennen, um eine Sicherheitskultur im Unternehmen zu entwickeln [16]. Die Verbesserung des Risikobewusstseins bei den Anwendern minimiert gleichzeitig benutzerspezifische Fehler und maximiert die Effektivität von Sicherheitsmaßnahmen.

Auf organisatorischer Ebene werden Programme auf Basis von Frameworks wie [17] zur Ausbildung von Benutzern genutzt. Erforderlich ist ebenso die technische Unterstützung zur Entwicklung eines Sicherheitsdenkens bei den Anwendern, wozu die Komponente "Security Awareness and Enforcement (SAE)" des MakoSi PIF zur interaktiven Risikovermittlung dient. Ziel ist die Entwicklung des Bewusstseins der Anwender für sicherheitsrelevante Fragestellungen bei der praktischen Arbeit [18].

Die SAE-Komponente (Abb. 6) dient als Schnittstelle von clientseitigen Anwendungen zu CIM/WBEM und stellt gleichzeitig die Enforcement- und Awareness-Informationen für Notifikatoren bereit. Die SAE-Komponente besteht aus mehreren Grundbausteinen: a) das CIM/WBEM-Plugin stellt die Schnittstelle zum PDP dar, b) der SAE-Service nimmt die zu prüfenden Aktionen von den Applikationen entgegen, mappt sie auf CIM-Events, leitet diese an das SAE-Plugin weiter gibt und das Ergebnis der Evaluierung der Aktion an die Applikation zurück und c) der Message-Service für rollen- und aktionsbasierte Meldungen für Notification-Clients. Die zu steuernde Applikation wird in diese Infrastruktur mittels Application Connector integriert. Die existierenden Applikationen können mittels Aspektorientierter Programmierung (AOP) eingebunden und um Funktionalität zur Sicherheitsprüfung erweitert werden. Für die Kommunikation zwischen den Applikationen und dem SAE Service bzw. den Notifikatoren (z.B. Monitor) und dem Message Service stehen verschiedene Protokolle, wie SOAP und Java RMI zur Verfügung.

Die SAE-Komponente ist proaktiv, sie sendet dem Benutzer via Message Service die Informationen, die als relevant für ihn eingeschätzt werden. Je nach Anwendungskontext können Alarme und visuelle Notifikationen in Abhängigkeit von den gemeldeten Ereignissen genutzt werden, um den Benutzer zu unterrichten. Dabei kann auch frei konfigurierbar vor möglichem Bedrohungspotenzial bei erlaubten Aktionen gewarnt werden.

8 Schlußfolgerungen

Unser Beitrag beschreibt die konkrete Ausprägung von bekannten Konzepten zur Umsetzung von Sicherheitspolitiken in der domänenübergreifenden Zusammenarbeit, mit speziellen Nebenbedingungen, die aus realitätsnahen, modernen Anwendungsszenarien erwachsen. Insbesondere tritt die Offenheit des Systems in Form durch Nutzer frei definierbarer, und von Nutzer zu Nutzer unterschiedlicher (heterogener) und oftmals asymmetrischer

Anforderungen an Politiken als komplizierender Faktor auf.

In der getroffenen, weitgehenden Abstraktion und der Beschränkung auf wenige, einfache Beispiel-Policies, rechtfertigt das betrachtete Beispiel-Szenario jedoch kaum den betriebenen Aufwand, insbesondere im Hinblick auf die verwendeten formalen Überprüfungsmethoden. In diesem Sinne verstehen wir unseren Beitrag als "proof of concept" einer allgemeinen Methode, die potenziell auch in komplexeren Fällen anwendbar zu sein verspricht — und je komplexer das betrachtete System, desto wichtiger werden formale Beweise unter klar beschriebenen Abstraktionsannahmen.

MakoSi zeigt auch die grundlegenden Schwierigkeiten eines umfassenden PIF auf, die kompletten, *a priori* gegebenen Rezepten und einer auch nur weitgehenden Automatisierung im Wege stehen. Der Anspruch auf Offenheit erschwert einerseits eine einheitliche Darstellungform für die Politiken, andererseits macht eine mächtige Policy-Beschreibungssprache aber formale Sicherheitsbeweise komplex bis zur Undurchführbarkeit. MakoSi versucht hier einen Mittelweg zu gehen mit der Verwendung einer ziemlich allgemeinen Policy-Beschreibung durch Event-Condition-Action Regeln, die aber auf einem sehr einfachen Zustandsmodell des Systems beruht. Die Aussagekraft erzielter formaler Beweise hängt letztlich auch von der Leistungsfähigkeit des Beweissystems (hier das SHVT) ab, die implizit die Abstraktion des Systemmodells mitbestimmt. Zumindest sind wir aber in der Lage, im Rahmen von MakoSi die getroffenen Abstraktionen, d.h. zunächst impizite Sicherheitsannahmen, explizit zu beschreiben und nachvollziehbar zu machen.

Literatur

[1] MakoSi Homepage. `www.makosi.de`

[2] V. Batra, J. Bhattacharya, H. Chauhan, A. Gupta, M. Mohania, U. Sharma. Policy Driven Data Administration. POLICY 2002, 220–223.

[3] David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn and Ramaswamy Chandramouli. Proposed NIST standard for role-based access control. *ACM Transactions on Information and System Security*, 10 no. 3, 2001.

[4] M. Covington and M. Moyer and M. Ahamad. Generalized role-based access control for securing future applications. 23rd National Information Systems Security Conference, Baltimore, MD, october 2001.

[5] Winfried Kühnhäuser. Metapolitiken, GMD Forschungszentrum für Informationstechnik, 1999, (GMD Research Series Nr. 13). Zugleich: Habil.-Schr. Univ. Chemnitz, 1999

[6] iViP Homepage `www.ivip.de`

[7] Uwe von Lukas. Cross-Enterprise Collaboration with iViP. In Concurrent Engineering: The Path to Electronic Commerce, SCS Publication, 2001.

[8] ISO/IEC standards ISO/IEC 17799:2000 Information security management: Codes of Practice; BS 7799-2:2002 Information security management: Specification with guidance for use.

[9] P. Ochsenschläger, J. Repp, R. Rieke, and U. Nitsche. The SH-Verification Tool - Abstraction-Based Verification of Co-operating Systems. *Formal Aspects of Computing*, 10:381–404, 1999.

[10] P. Ochsenschläger, J. Repp, and R. Rieke. Abstraction and composition - a verification method for co-operating systems. *J. Exp. Theo. Artificial Intelligence*, 12:447–459, 2000.

[11] P. Ochsenschläger, J. Repp, and R. Rieke. The SH-Verification Tool Proc. 13th International Florida Artificial Intelligence Research Society Conference (FLAIRS), 2000.

[12] DMTF Web-Based Enterprise Management Initiative.
www.dmtf.org/standards/standard_wbem.php

[13] DMTF Draft DSP0108: CIM Core Policy Model White Paper.
www.dmtf.org/standards/documents/CIM/DSP0108.pdf

[14] B. Moore, E. Ellesson, J. Strassner, A. Westerinen. Policy Core Information Model.
www.ietf.org/rfc/rfc3060.txt

[15] DMTF Draft DSP0201: Representation of CIM in XML.
www.dmtf.org/standards/documents/WBEM/DSP201.html

[16] Fowler J. Developing The Security Culture At The SEISMED Reference Centres. In: B. Barber et al, editors. Towards Security in Medical Telematics: Legal and Technical Aspects. IOS Press, Amsterdam, pp156-161, 1996.

[17] NIST Information Technology Security Training Requirements: A Role-and Performance-Based Model (supersedes NIST Spec. Pub.500-172), SP 800-16, March 1998.

[18] Marc Welz and Andrew Hutchinson. Interfacing Trusted Applications with Intrusion Detection Systems, RAID 2001.