Werkzeuggestützte Identifikation von IT-Sicherheitsrisiken in Geschäftsprozessmodellen

Marc Peschke (Softlution Deutschland GmbH), Martin Hirsch (FH Dortmund), Jan Jürjens (TU Dortmund und Fraunhofer ISST), Stephan Braun (TU Dortmund)

Website: http://jan.jurjens.de

Zusammenfassung

In diesem Artikel wird ein neu entwickeltes Werkzeug vorgestellt, mit dem IT-Sicherheits-Risiken in Geschäftsprozessmodellen identifiziert werden können.

Einleitung

Heutzutage werden immer mehr elektronische Daten und Informationen produziert, verarbeitet und gespeichert. Diese Daten sind von sehr großem Wert für Unternehmen und Behörden und es gilt diese, sowie auch deren Austausch, angemessen zu schützen. Die verteilte Haltung und Mobilität der Daten und Informationen erfordert einen Schutz der neu entstandenen Kommunikationswege. Sicherheitsbedrohungen sind durch externe und auch interne Angreifer ständig präsent, wodurch immer wieder neue Angreifer-Szenarien entstehen, deren Auswirkungen bei Design und Entwicklung einer Software noch nicht bekannt waren. Um solche Schwachstellen im Vorfeld aufzudecken und die Sicherheit aufrecht zu erhalten, wird Risikomanagement in der IT-Sicherheit immer wichtiger.

Innovative Methoden der Softwareentwicklung, wie zum Beispiel der modellgetriebene Softwareentwicklungsansatz, bieten die Möglichkeit, Sicherheitsaspekte eines Systems bereits in frühen Phasen in die Spezifikationen mit einfließen zu lassen. Basierend darauf werden Sicherheitsanalysen durchgeführt, wodurch Schwachstellen schon während der Entwicklung an identifiziert und behoben werden können.

Das vorliegende Papier stellt ein Werkzeug auf Basis der UMLsec-Werkzeugumgebung UMLs11] vor, mit dem IT-Sicherheitsrisiken in Geschäftsprozessmodellen auf Basis von UML-Aktivitätsdiagrammen identifiziert werden können. Wir demonstrieren das Werkzeug anhand eines Beispielszenarios aus dem Bereich Cloud-Computing.

Verwandte Arbeiten

Im Bereich der modellgetriebenen Softwareentwicklung existieren bereits einige Ansätze, die Sicherheitsanforderungen an die Software berücksichtigen.

Das Ziel des SecReq Projektes [KnSJ09] ist es, eine Unterstützung in allen Schritten einer Sicherheitsanforderungserhebung zu bieten. Es soll Mechanismen bieten, um aus textuellen Sicherheitsanweisungen Sicherheitsanforderungen zu extrahieren. Dies soll dazu führen, die Lücke zwischen den bewährten Methoden und der Erfahrungen von Programmieren und Designern zu verkleinern. SecReq kombiniert drei charakteristische Techniken, um dieses Ziel zu erreichen. Als Erste ist die Common Criteria zu nennen [ISO07], ein Katalog und Anleitung zur Umsetzung von Teilen der Kriterien in IT-Sicherheitsspezifikationen. Das HeRa Werkzeug [Knau09] bietet sicherheitsrelevante Heuristik-Regeln und des Weiteren wird das UMLsec Werkzeug [Jürj05] für die Sicherheitsanalyse und das Sicherheitsdesign verwendet. SecReq analysiert einen Text auf sicherheitsrelevante Worte und liefert eine Wahrscheinlichkeit, wie stark das jeweilige Wort als Sicherheitswort eingestuft werden kann.
Octave [DoWo03] ist ein risikobasiertes Abschätzung- und Planungswerkzeug für Strategien im Bereich IT-Sicherheit. Octave hat anders als andere Ansätze den Fokus auf organisatorische und praxisrelevante Sicherheitsbelange. Der Octave Ansatz ist motiviert durch zwei grundsätzliche Aspekte. Zum einen betriebsbedingte Risiken und zum anderen gängige Sicherheitspraktiken. Octave bietet einem Unternehmen die Möglichkeit Sicherheitsbelange aufzuspüren. Dies geschieht in drei Phasen. In Phase 1 wird ein Gefahrenprofil erstellt. Aufgrund dieses Profils wird nun in Phase 2 versucht, Verwundbarkeiten in der Infrastruktur zu identifizieren. Letztendlich wird in der dritten Phase eine Sicherheitsstrategie und ein Sicherheitsplan entwickelt [DoWo03].
Coras [Gran02] bietet ein Framework an, mit dem es möglich ist, präzise, eindeutige und effiziente Risikoanalysen von kritischen Sicherheitssystemen zu erstellen. Hierbei wird besonderes Augenmerk auf die Möglichkeiten der Modellierung in UML gelegt. Coras soll Hilfestellung bei typischen Sicherheitsfragen geben, speziell aber im Bereich der IT-Sicherheit. Coras integriert alle Aspekte bezüglich der Definition, des Erreichen und des Aufrechterhaltens von Vertraulichkeit, Integrität, Verfügbarkeit, Authentizität und Verlässlichkeit. Zudem bezieht Coras die menschliche Interaktion in die technologischen Aspekte mit ein [Gran02].
Die Arbeit von Fenz, Ekelhart, Neubauer [FeEN09a] befasst sich mit der Risikoerfassung von Prozessen. Es wird eine Wahrscheinlichkeit für eine gegebene Gefahrenquelle zu einer speziellen potentiellen Verwundbarkeit ermittelt.
Mit dem GSTool [Hump03] stellt das Bundesamt für Sicherheit in der Informationstechnik ein Werkzeug zur Verfügung, mit dem das Erstellen und Verwalten von Sicherheitskonzepten entsprechend dem IT-Grundschutz [BSI10] unterstützt wird und dem Benutzer die Anwendung dieser Konzepte erleichtert. Die Erfahrungen und Ergebnisse dieser Arbeit können durch ein Webinterface benutzt werden. Die Grundlage dieses Webinterfaces ist die Security Ontology, die im Rahmen dieses Projektes entwickelt worden ist [FeEN09b].

Die aufgeführten Ansätze bieten jedes für sich Vor- und Nachteile. Bei bisherigen Werkzeugen zur automatisierten Risikoanalyse ist es oftmals nötig, viele, nur für den Fachmann, verständliche Informationen bzw. Daten zusammen zu tragen und an das Werkzeug zu übergeben. Die Zusammenstellung dieser Datenmenge ist in der Regel sehr umständlich und zeitaufwendig. Die Vorgehensweise zur Erhebung dieser Daten verlangt beispielsweise ein Aufstellen eines Netzplanes des Unternehmens. Zudem müssen alle Anwendungen erfasst werden, die dort zum Einsatz kommen. Des Weiteren ist es notwendig, alle Kommunikationsschnittstellen im Unternehmen zu bestimmen. Die Java-Komponente, die im Rahmen dieses Artikels entwickelt wurde, benötigt eine solch aufwendige Datenerhebung nicht. Die Komponente arbeitet allein auf den Daten, die aus dem Aktivitätsdiagramm gewonnen werden können, in dem ein Geschäftsprozess modelliert worden ist.

Werkzeuggestützte Identifikation von IT-Sicherheitsrisiken

Das in diesem Artikel vorgestellte Werkzeug ist in Form eines Plugins - dem ’Riskfinder’- zum Auffinden risikobehafteter Aktivitäten in einem Geschäftsprozess, für das bereits bestehende UMLsec-Fra-

mework [UMLs11] implementiert worden. Dieses Framework wird in diesem Abschnitt in seinen Grundzügen vorgestellt. Zusätzlich werden die Überlegungen aufgezeigt, die im Vorfeld der Implementation des Plugins notwendig waren.

Das UMLsec-Framework [UMLs11,HöJü08,PWXG+07,JüSc08,JüSh07] bietet dem Benutzer die Möglichkeit, Modelle in der Unified Modeling Language (UML), die gemäß der UML-Sicherheitserweiterung UMLsec mit sicherheitsrelevanten Informationen annotiert wurden, auf Einhaltung der annotierten Sicherheitsanforderungen zu überprüfen. Der Funktionsumfang des Frameworks lässt sich beliebig durch Plugins erweitern, die in die Struktur des Programmes eingegliedert werden. Abbildung 3.1 zeigt eine Darstellung des UMLsec-Frameworks, die den Workflow während der Analyse darstellt. Der Benutzer lädt ein UML-Modell in das UMLsec-Framework, das er zuvor mit einem UML-Editor erstellt hat. Das Modell wird nun in einen MDR-Container transferiert, der dem Analyseplugin zur Verfügung steht. Die Daten des MDR-Container werden nun in den jeweiligen Analysebausteinen bearbeitet und die Ergebnisse dann als Text ausgegeben.

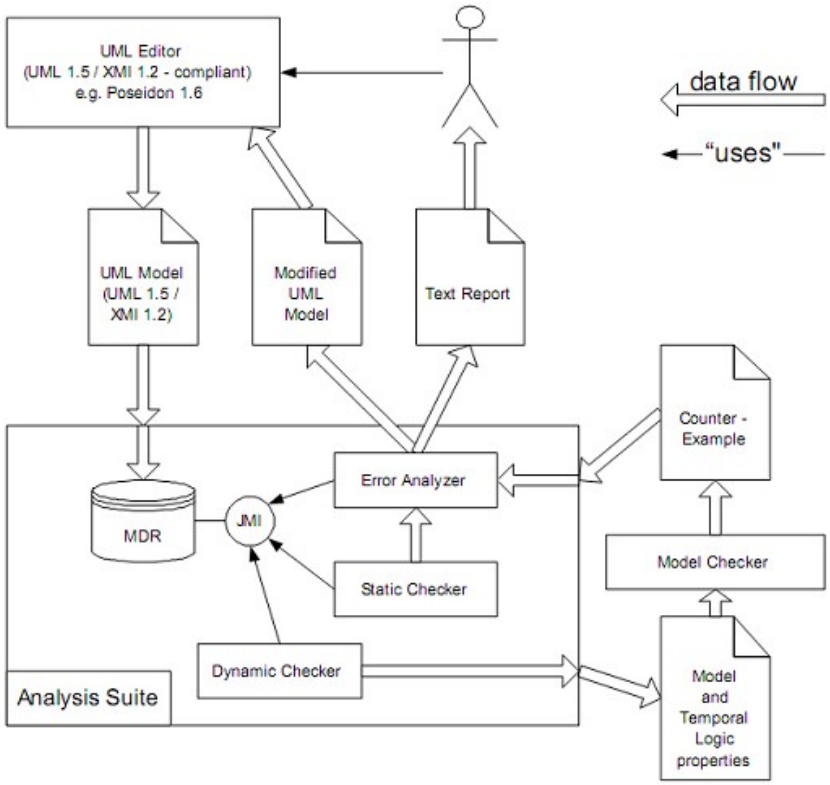


Abb. 3.1: UMLsec-Framework

Im Vorfeld der Entwicklung des Riskfinder-Plugins wurden Anforderungen an die Bedienbarkeit und Funktion festgelegt. Ziel war eine prototypische Entwicklung eines Plugins für das UMLsec-Framework, das dem Benutzer eine Unterstützung zur automatisierten Risikoanalyse von Geschäftsprozessen bietet. Dafür war es notwendig, die in Aktivitätsdiagrammen modellierten Geschäftsprozesse in das UMLsec-Framework einzulesen und Informationen auszulesen. Hierfür wurde die bereits bestehende Schnittstelle des UMLsec-Frameworks genutzt und geeignet erweitert. Bisher war es möglich UML-Diagramme einzulesen. Die Manipulation dieser Diagramme sollte nun ermöglicht werden, um die Ergebnisse der Analyse für den Benutzer sichtbar zu machen. Für die eigentliche Analyse ist es notwendig eine Grundlage an Regeln zu definieren, aufgrund derer das Auffinden von Sicherheitsrisiken möglich ist und für den Benutzer eine Lösungsfindung erleichtert wird. Dazu soll ein Sicherheits-Repository entwickelt werden, das als Datenstruktur dient und die Sicherheitsregeln in geeigneter

Form abspeichert. Dieses Sicherheits-Repository soll transparent für den Benutzer sein. Es soll ihm möglich sein, das Sicherheits-Repository zu erweitern und zu modifizieren.
Für die Entwicklung des Plugins war es nötig, zuerst einen Plan aufzustellen, welche Funktionen wie umgesetzt werden können, um einen Zugriff der einzelnen Modulteile untereinander zu gewährleisten. Das Riskfinder-Plugin benötigt Zugriff auf die eingelesene ArgoUML-Datei, da das Riskfinder-Plugin die risikobehafteten Aktivitäten farblich kennzeichnen muss und auch eine Notiz einfügen soll. Somit ist der Zugriff auf die Daten und die Weiterverarbeitung notwendig. Zudem wurden einige Veränderungen an der GUI des UMLsec-Frameworks vorgenommen, die es ermöglichen, Manipulationen an dem Sicherheits-Repository vornehmen zu können. Die Schnittstelle zwischen dem Riskfinder-Plugin und dem Sicherheits-Repository erfordert es, das Repository einzulesen. Das Sicherheits-Repository liegt im XML-Format vor und kann mit einem XML-Parser eingelesen werden. Zudem soll es ermöglicht werden, auf das Sicherheits-Repository zuzugreifen, um zum einen neue Datensätze hinzuzufügen und zum anderen auch veraltete Datensätze zu löschen. Diese Veränderungen ermöglichen erstmalig einen interaktiven Umgang des UMLsec-Frameworks mit den UML-Diagrammen. Zuvor war es nur möglich, Diagramme einzulesen und eine Textausgabe zu studieren. Nun agiert das UMLsec-Framework aktiv und verändert Diagramme gemäß den Ergebnissen der Analysen.

Die Klasse ’Riskfinder’ bildet die Ausgangsklasse des Plugins. Diese wird in das UMLsec-Framework eingebunden und regelt die Zugriffe des Frameworks auf das Modul. Sie greift auf die Klasse ’CheckerRiskfinder’ zu, in der der Analysealgorithmus implementiert ist. Diese Klasse greift beim Abarbeiten des Algorithmus auf die Helper-Klassen zu. Diese bieten Funktionalitäten, mit denen XML-Dateien eingelesen, gespeichert und modifiziert werden können und werden zum Beispiel benötigt, um die Layout-Datei der ArgoUML-Datei zu bearbeiten.

Umsetzung

In diesem Abschnitt wird die Umsetzung der Überlegungen dargestellt. Hierfür werden zunächst die Komponenten des Riskfinder-Plugins dargestellt, die für die Verarbeitung und den Austausch von Daten mit anderen Teilen des Programms implementiert werden mussten.

Zuerst ist eine Schnittstelle nötig, die das Einlesen und Bearbeiten einer Zargo-Datei ermöglicht. Diese ist für die Analyse durch das Riskfinder-Plugin notwendig. Zudem ist eine Schnittstelle zum Sicherheits-Repository nötig. Diese erlaubt den Zugriff auf das Sicherheits-Repository durch das Riskfinder-Plugin. Zusätzlich wird eine Schnittstelle eingebaut, die den Zugriff auf die Daten und Informationen des IT-Grundschutz ermöglicht, mit denen das grundlegende Sicherheitskonzept in Form von Mustern der Sicherheitsanforderungen, den Sicherheits-Pattern, erstellt werden. Das UMLsec-Framework bietet den Rumpf für das Riskfinder-Plugin. Zudem war es nötig, einige Veränderungen in dem UMLsec-Framework vorzunehmen, die es ermöglichen, auf die ArgoUML-Datei zugreifen zu können. Der Riskfinder benötigt diesen Zugriff, um die Ergebnisse der Analyse in das Diagramm einzufügen. Ein weiterer Punkt, der in das UMLsec-Framework implementiert worden ist, ist der Menüpunkt ’Security Repository’. Abbildung 4.1 zeigt das UMLsec-Framework mit dem neuen Menüpunkt.

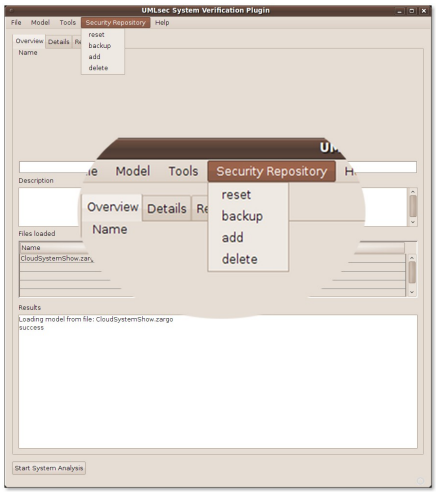


Abb. 4.1: UMLsec-Framework mit Repository-Menü

Die Logik hinter diesen Knöpfen steckt in der Klasse ’GuiHelper’ des Riskfinder-Plugins. Es werden hier die Punkte ’reset’, ’backup, ’add’ und ’delete’ angeboten.’Reset’ leert das Sicherheits-Repository, nachdem ein Backup in das Home-Verzeichnis des Benutzers angelegt wird. ’Backup’ macht ein Backup ins Home-Verzeichnis des Benutzers. Für die Punkte ’add’ und ’delete’ wird jeweils ein neues Fenster aufgebaut. Für die ’delete’-Aktion werden alle bereits vorhandenen Sicherheits-Pattern in einer Liste angezeigt, aus der das zu löschende Pattern ausgewählt werden kann. Bild 4.2 stellt die GUI zum Löschen eines Repository-Pattern dar.

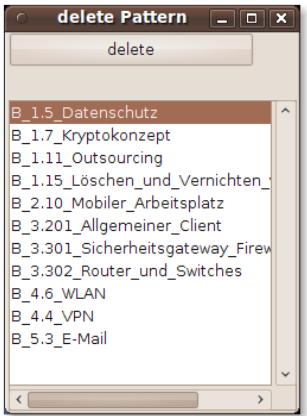


Abb. 4.2 GUI zum Löschen eines Sicherheit-Pattern

Beim Aufruf von ’add’ wird ein Fenster geöffnet, das mit wenigen Klicks ein Sicherheits-Pattern erstellt und in das Sicherheits-Repository einfügt. Neben dem Namen und dem Bereich, wird eine Liste für jeden Teil des Gefahren-und Maßnahmenkataloges des IT Grundschutzes angeboten, aus denen die entsprechenden Punkte ausgewählt und per Klick auf die Überschrift der jeweiligen Spalte eingeloggt werden können.
Die Struktur des Sicherheitsrepositories leitet sich vom Aufbau des IT-Grundschutz des Bundesamt für Sicherheit in der Informationstechnik ab. Die einzelnen Punkte des Bausteinkataloges bilden die jeweils nächste Ebene des Repositories.

Die letzte Ebene bilden die Gefahren und Maßnahmen des jeweiligen Bausteins. Für die Implementierung des Algorithmus wurden verschiedene Ansätze betrachtet. Der erste Ansatz bestand darin, Vorgaben für die Aktivitätennamen anzubieten, die sich auf das Grundvokabular des IT-Grundschutzes beziehen. Hierbei werden dem Benutzer Begriffe angeboten, die er aus einer Liste übernehmen kann. Diese Begriffe kann er sich nach dem Bausteinprinzip zusammen klicken und damit seine Pro-

Abb. 4.4: Baustein G und M

zesse benennen. Dieser Ansatz ist für den Benutzer unpraktisch und zeitaufwendig. Er ist daran gebunden, was ihm die Auswahlliste vorgibt. So kann es zum Beispiel vorkommen, dass aus verschiedenen Gründen kein passender Baustein zur Verfügung steht. Der Benutzer soll nicht eingeschränkt werden. Es soll möglich sein, dass man seine Prozesse frei modellieren kann, ohne sich von einer Vorauswahl einschränken zu lassen. Daher wurde diese Methode verworfen. Alternativ wurden Stoppwörter benutzt. Stoppwörter werden in Volltextindizierungen nicht beachtet. Sie treten sehr häufig auf und haben im Allgemeinen keine Relevanz für den Inhalt des Dokuments, da sie überwiegend grammatikalische und syntaktische Funktionen übernehmen. Aus diesem Grund werden diese Stoppwörter bei dem später durchzuführenden Vergleich nicht benötigt. Sie würden das Ergebnis verfälschen und zu einer zu großen Trefferzahl führen. Ein weiteres Hilfsmittel die Menge der zu untersuchenden Wörter zu verkleinern und nur mit hilfreichen Worten zu füllen, ist das SecReq-Projekt der Universität Hannover. Im Rahmen dieses Projektes wurde ein Tool entwickelt, das mit heuristischen Algorithmen Worte eines Textes auf ihre Sicherheitsrelevanz untersucht. Dafür werden textuelle Sicherheitsanforderungen in das Tool geladen. Diese untersucht das Werkzeug und gibt eine Liste von Worten, einen Wortvektor, aus. Jedes Wort erhält einen Wert, der aussagt wie relevant dieses Wort ist.

Der Algorithmus, der im Rahmen dieses Artikels implementiert worden ist, arbeitet auf Basis von Wortvektoren. Diese werden nach bestimmten Kriterien erstellt und verglichen. Abbildung 4.3 verdeutlicht das Grundprinzip des Algorithmus. Als Eingabe wird ein Aktivitätenname übergeben und eine Empfehlung auf Basis des IT-Grundschutzes als Ausgabe generiert. In Schritt 1 werden die Sicherheits-Pattern-Objekte erzeugt. Die nötigen Informationen werden aus dem Sicherheits-Repository geladen und die auf diese Weise identifizierten Wörter in einen Vektor geschrieben.

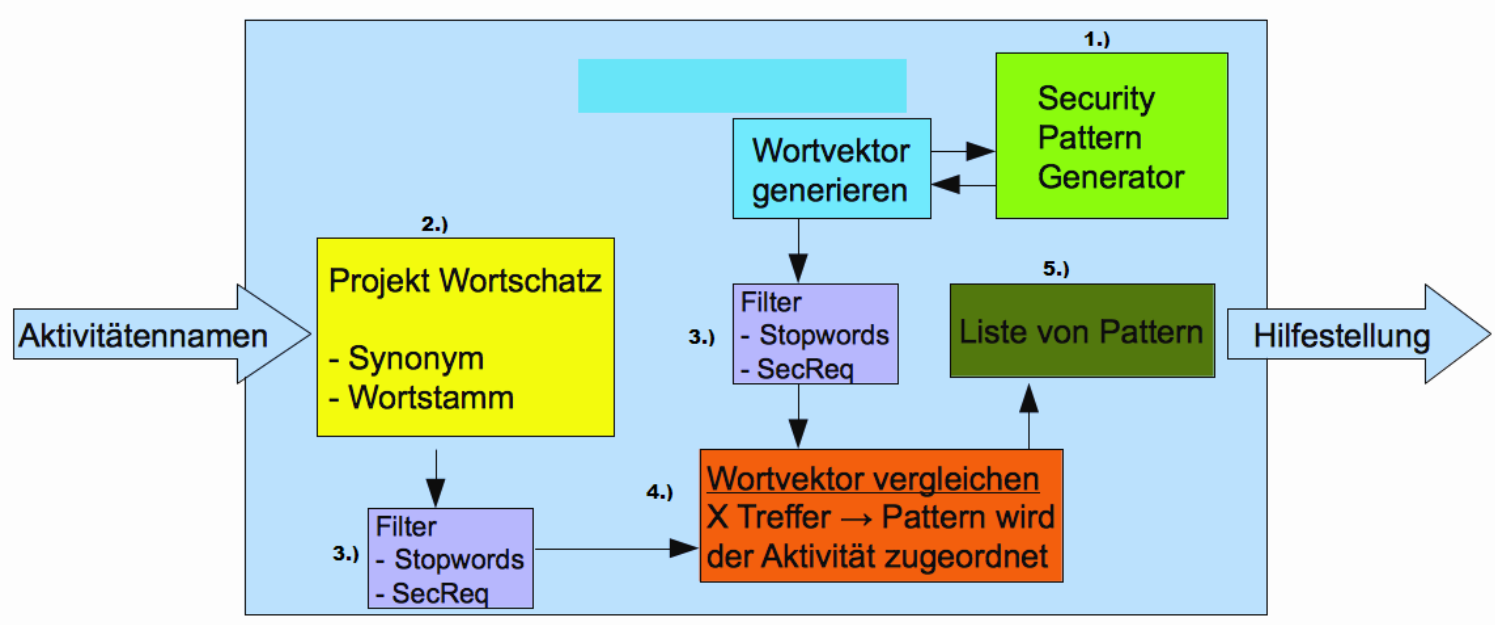


Abb. 4.3: Riskfinder-Algorithmus

Im zweiten Schritt werden die Methoden des Projekts “Wortschatz” der Universität Leipzig angewendet: Die Aktivitätennamen werden als Eingabe übergeben. Man erhält nun für jedes Wort eine Liste von Synonymen und Worten die mit dem Ausgangswort mit einer bestimmten Varianz in Verbindung stehen. Dieses wird ausgeführt, um den Wortvektor der Eingabe, der im ungünstigsten Fall aus nur einem Wort besteht, zu vergrößern. Dieses vermeintliche Aufblähen bewirkt allerdings eine höhere Anzahl an Treffern beim späteren Vergleich. In Schritt 3 werden nun Filter auf die Synonyme und die Wortvektoren angesetzt. Die Wortvektoren werden durch Anwendung zweier Methoden gefiltert.

Hierzu werden a) Stoppwörter eliminiert und b) die Worte des Vektors untersucht und nach ihrer Sicherheitsrelevanz eingestuft. Hierdurch wird sichergestellt, dass nur Worte in den Wortvektoren enthalten sind, die eine gewisse Sicherheitsrelevanz besitzen. In Schritt 4 geschieht der eigentliche Wortvergleich auf Basis der beiden Wortvektoren. Aus dem Wortvektor der Aktivität werden zunächst die Stoppworte herausgefiltert. Dann werden die Worte des Vektors, mit denen des Sicherheits-Patterns verglichen. Wenn nun das Limit für die Anzahl der Treffer erreicht worden ist, wird das aktuelle Pattern der aktuellen Aktivität zugeordnet. Es kann also davon ausgegangen werden, dass das Pattern einen Aspekt der Sicherheit enthält, der für diese Aktivität relevant ist und dem Benutzer helfen kann, seine Modellierung zu überarbeiten. In Schritt 5 wird eine Liste für die Ausgabe aufbereitet. In dieser werden zu jeder Aktivität, zu der ein Sicherheits-Pattern identifiziert worden ist, diese aufgelistet. Ist eine Aktivität mehr als einem Pattern zugeordnet worden, so wurde diese mehrmals in den Vektor ’patternFounds’ eingetragen. Die Liste wird dann als Übersicht in das Diagramm eingefügt und die jeweiligen Aktivitäten werden je nach Einstufung beziehungsweise Auftauchen in dem Vektor ’patternFounds’ in den Farben gelb, orange oder rot eingefärbt

Anwendung

Die Implementierung wurde mittels einer Fallstudie getestet. Diese beschäftigt sich mit einem Anwendungsbeispiel, an dem die Funktion des Riskfinder-Plugins und die Resultate der Analyse gezeigt werden. Als Beispiele wurde ein Szenario ausgewählt und am Ende wird eine Evaluation durchgeführt, die zeigen soll, wie die Qualität des Wortvektorvergleichs ist. Abbildung 5.1 zeigt, wie dieses Szenario in einem Prozess modelliert werden kann. Der Prozess beginnt mit der Aktivität ’Kunde benötigt Cloud-Service’. Die Anfrage des Kunden an den Dienstleister des Cloud-Service wird in der Aktivität ’Dienstleister erhält Serviceanfrage‘ modelliert.

5.1 Beispielszenario: Cloud-Dienstleistung

Dieses Beispiel beschreibt einen Geschäftsprozess aus dem Bereich des Cloud-Computing. In diesem Beispiel benötigt ein Kunde einen Dienst im Internet, den er bei einem Cloud-Anbieter angeboten bekommt. Um sich zu authentisieren, verlangt der Cloud-Anbieter die Daten des Kunden. Werden diese Daten nach einer Prüfung validiert, so wird der Service dem Kunden gewährt und ihm eine Abrechnungsemail zugesandt. Sollte die Prüfung negativ verlaufen sein, wird dem vermeintlichen Kunden die Dienstleistung nicht gewährt.

Abb. 5.2 Ausgabe des Riskfinders

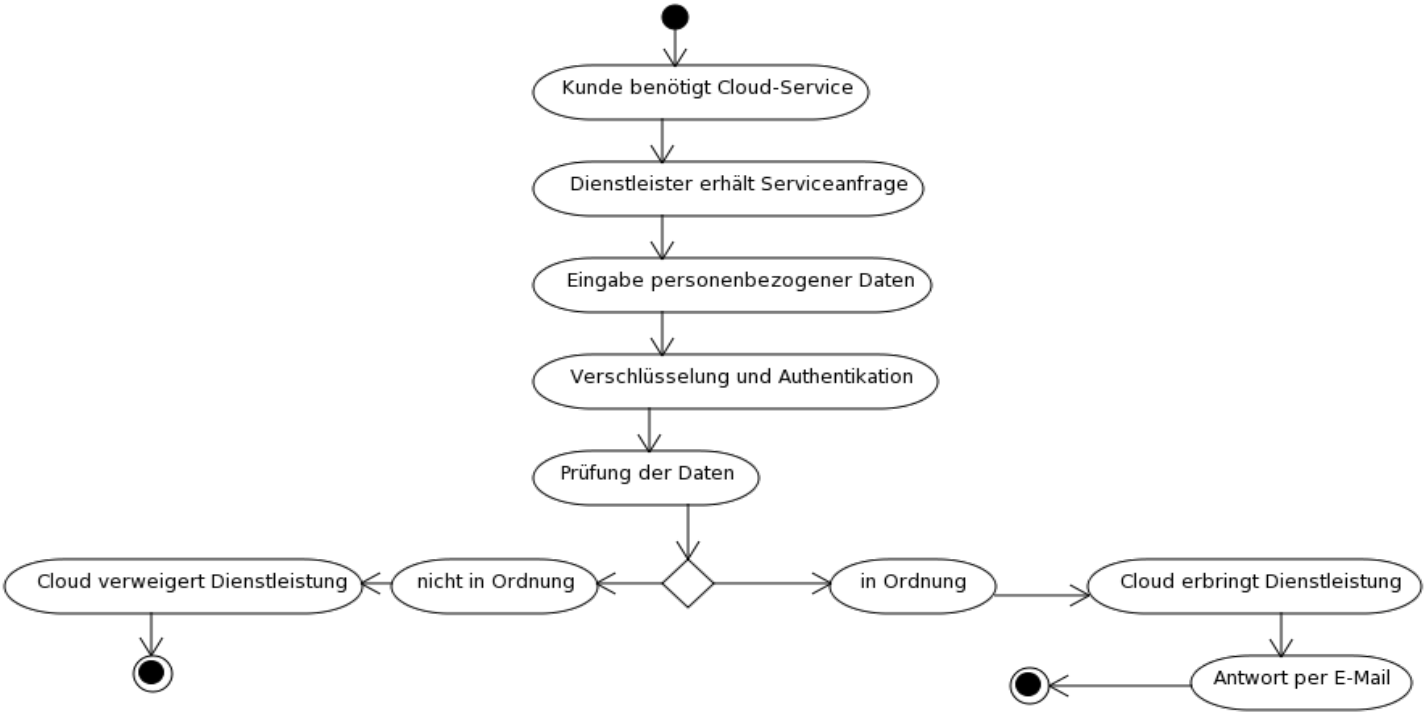
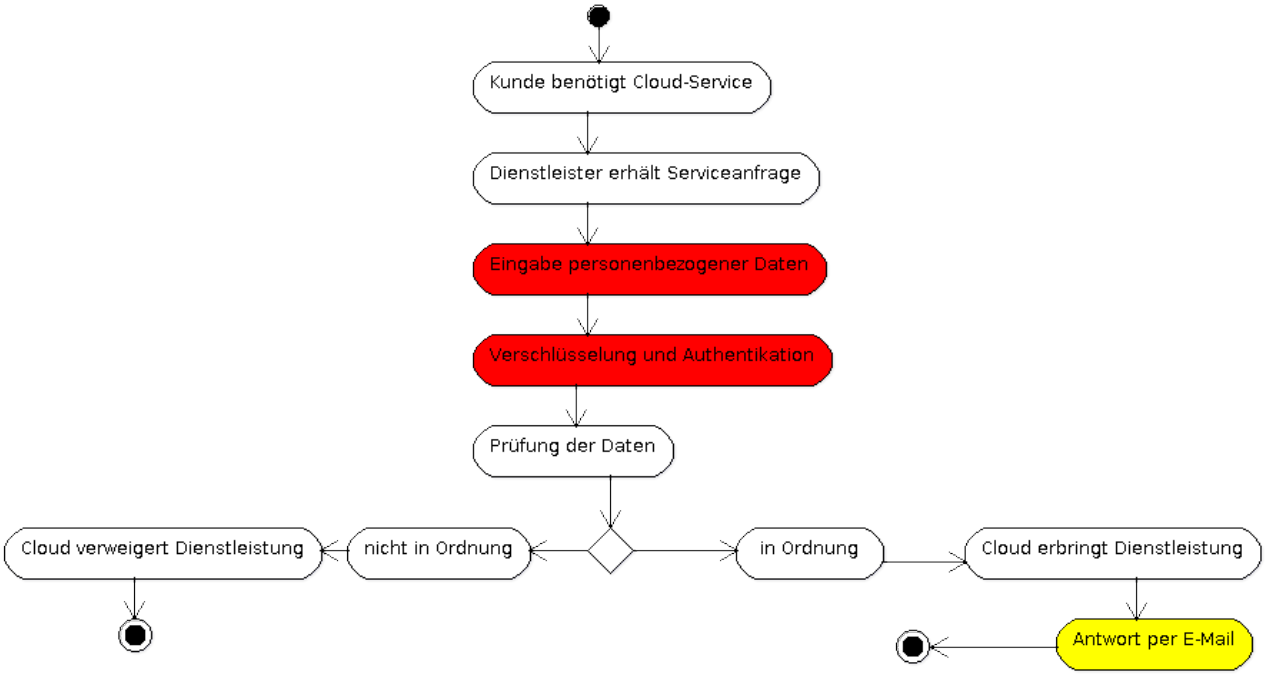


Abb. 5.1: Das Beispielszenario

Der Kunde muss seine Daten dem Dienstleister mitteilen. Dies geschieht in der Aktivität ’Eingabe personenbezogener Daten’. Diese Daten werden nun verschlüsselt und an den Dienstleister übersendet. Aktivität ’Verschlüsselung und Authentifikation’ beschreibt den Vorgang der Übermittlung der verschlüsselten Daten. Nun werden die personenbezogenen Daten des Kunden in der Aktivität ’Prüfung der Daten’ überprüft. Sind diese Daten nicht in Ordnung, so setzt sich der Prozess mit den Aktivitäten ’nicht in Ordnung’ und ’Cloud verweigert Dienstleistung’ fort und der Dienstleister. verweigert dem Kunden seine Dienste. Sind die Daten in Ordnung, wird in der Aktivität ’Cloud erbringt Dienstleistung’ dem Kunde seine Anfrage erfüllt und ihm ein Dienst frei geschalten. Am Ende wird eine E-Mail mit einer Bestätigung und Abrechnung an den Kunden versendet. Dies ist modelliert in der Aktivität ’Antwort per E-Mail’. Dieses Modell ist in Form eines Aktivitätsdiagramms umgesetzt und kann so später für die Analyse mit dem UMLsec-Framework benutzt werden.

Nach Ausführung der Analyse durch das Riskfinder-Plugin des UMLsec-Werkzeugs sieht das Diagramm aus, wie in Bild 5.2 zu sehen. Man erkennt die Aktivitäten, die eingefärbt und in der Notiz vermerkt wurden. In der Notiz sind die Sicherheits-Pattern aufgelistet, bei dem der Riskfinder-Algorithmus einen Risikoverdacht geäußert hat.




```
Risikoverdacht bei [Eingabe personenbezogener Daten]:
1 | B_1.5_Datenschutz
2 | B_1.7_Kryptokonzept
3 | B_1.11_Outsourcing
4 | B_1.15_Löschen_und_Vernichten_von_Daten
5 | B_2.10_Mobiler_Arbeitsplatz
6 | B_3.201_Allgemeiner_Client
7 | B_3.301_Sicherheitsgateway_Firewall
8 | B_3.302_Router_und_Switches
9 | B_4.6_WLAN
10 | B_4.4_VPN
11 | B_5.3_E-Mail
---
Risikoverdacht bei [Antwort per E-Mail]:
1 | B_5.3_E-Mail
---
Risikoverdacht bei [Verschlüsselung und Authentikation]:
1 | B_1.7_Kryptokonzept
2 | B_1.7_Kryptokonzept
3 | B_1.11_Outsourcing
4 | B_1.11_Outsourcing
5 | B_1.15_Löschen_und_Vernichten_von_Daten
6 | B_1.15_Löschen_und_Vernichten_von_Daten
7 | B_2.10_Mobiler_Arbeitsplatz
8 | B_2.10_Mobiler_Arbeitsplatz
9 | B_3.201_Allgemeiner_Client
10 | B_3.201_Allgemeiner_Client
11 | B_3.301_Sicherheitsgateway_Firewall
12 | B_3.301_Sicherheitsgateway_Firewall
13 | B_3.302_Router_und_Switches
14 | B_3.302_Router_und_Switches
15 | B_4.6_WLAN
16 | B_4.6_WLAN
17 | B_4.4_VPN
18 | B_4.4_VPN
19 | B_5.3_E-Mail
20 | B_5.3_E-Mail
---
```

Abb. 5.2: Ausgabe des Riskfinders

Evaluierung

Die Evaluierung zeigt, dass der Analysealgorithmus des Riskfinder-Plugins eine signifikante Trefferanzahl für Sicherheitsbezogene Texte erlangt. Dazu werden drei Texte in das Riskfinder-Plugin

geladen und analysiert. Der erste Text hat keine Inhalte, die sich nicht mit sicherheitsrelevanten Themen beschäftigen. Bei dem Text handelt es sich um einen Auszug aus dem Märchen 'Aschenputtel' der Gebrüder Grimm [Grim03] und ist als nicht sicherheitsrelevanter Text ideal. Als zweiten Text ziehen wir einen Text heran, der das Thema IT-Sicherheit thematisiert. Dabei handelt es sich um 'Vom Internet der Computer zum Internet der Dinge' [MaFl10]. Der dritte Vergleichstext wird ein Text mit sicherheitsrelevanten Themen aus dem Grundschutzkatalog heran genommen und analysiert. Dabei handelt es sich um einen Artikel des BSI, der sich mit der Verschlüsselung von Daten befasst [BSI10b]. Dies ist daher notwendig, um den Gebrauch des Grundvokabulars des IT-Grundschutz zu simulieren. Es sollte nun so sein, dass der Algorithmus bei dem nicht-IT-verwandten-Text weniger Treffer erzielt, als bei dem IT-Text.

Abbildung 6.1 zeigt die Ausgabe des Validierungsprozesses. Zu sehen ist, dass sich der Test in drei Abschnitte gliedert. Zum einen wurde der normale Text ohne Sicherheitsrelevanz untersucht. Das Ergebnis zeigt die Anzahl der Treffer für eine Analyse mit und ohne Einsatz des Projektes "Wortschatz". In dem normalen Text hat der Algorithmus einen Treffer mit Wortschatz-Webservice und keinen Treffer ohne Wortschatz-Webservice erzielt. Dies ist ein gutes Ergebnis, da für diesen Text ein Ergebnis von Null ein optimales Ergebnis darstellt. Zum anderen wurde der normale Text untersucht, der sich mit einem IT-Sicherheitsthema befasst. Es handelt sich um ein Paper über die Veränderung des Internets [MaFl10], einmal mit und einmal ohne Einsatz des Projektes "Wortschatz". Hier zeigt das Ergebnis, das in Abbildung 6.1 dargestellt ist, dass sowohl mit und ohne das Projekt "Wortschatz" elf Treffer erzielt worden sind. Somit ist es wahrscheinlich, dass der Text als ein sicherheitsrelevanter Text eingestuft werden kann. Das dritte Ergebnis bezieht sich auf einen stark sicherheitsrelevanten Text aus dem 'BSI-Standard' [BSI10a]. Hier findet der Algorithmus 9 Treffer. Das dritte Beispiel ist ein Text, der das Vokabular des IT-Grundschutzes verwendet. Für diesen Fall ist eine besonders gute Trefferrate zu erwarten und der Algorithmus des Riskfinder-Plugins arbeitet besonders gut.

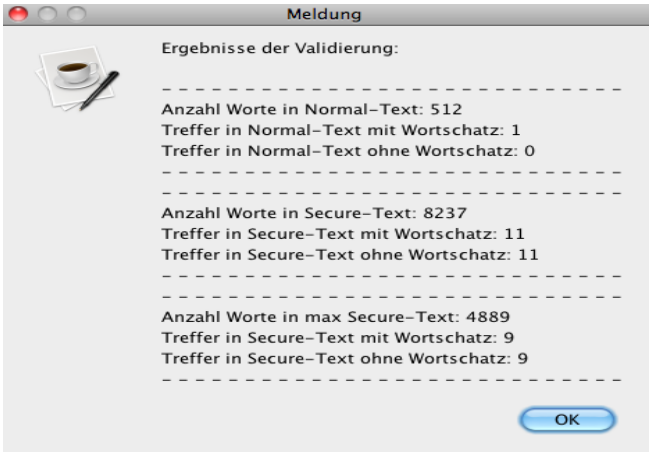


Abb. 6.1: Ergebnisse der Validierung

Für eine bessere Einordnung der Ergebnisse setzen wir die Anzahl der Treffer in Relation zu der Anzahl der Worte des jeweiligen Textes. Damit setzen wir die Ergebnisse direkt mit Prozentwerten in Verbindung. Tabelle 6.2 zeigt eine Auflistung der Werte die sich aus den Ergebnissen ergeben. Man sieht, dass sich die Treffer pro 100 Worte je nach Text erhöhen. Es wurden immer nur Auszüge aus den jeweiligen Dokumenten analysiert, da durch den Einsatz der Webservices des Projektes "Wortschatz" die Laufzeit pro Wort stark wächst. Aber auch so ist schon ein signifikantes Ergebnis zu erkennen. Dieses Ergebnis muss nun noch in einen globalen Kontext gebracht werden. Dafür ist es

notwendig einen kurzen Exkurs in den deutschen Satzbau zu machen. Ein Satz in der deutschen Sprache besteht im Grunde aus fünf Bestandteilen [ArZy02].

- Subjekt: Gegenstandswort wie ’Tisch. Begriffswort wie ’Mut’.
- Prädikat: Verb (Tu-Wort) wie laufen, springen
- Adjektiv: Wie-Wort. groß, klein
- Konjunktion: Verbindungwort von Sätzen. weil, um
- Pronomen: für-Wort. der, dieses
- Adverb: wie ein Adjektiv. Bezug auf das Verb.
- (un-) bestimmte Artikel: der, die, das

Die Syntax der deutschen Sprache für einen normalen Satz zeigt der folgende Beispielsatz. ’Der junge Mann läuft schnell.’ Dieser Satz enthält fünf der sieben oben aufgelisteten Bestandteile. Wenn wir nun davon ausgehen, dass diese Bestandteile immer in einem normalen Satz enthalten sind, so ergibt das einen Anteil von 20% pro Wortart. Da ein wissenschaftlicher Text aus komplexen Sätzen besteht, die auch Nebensätze und somit auch Konjunktionen enthalten, verändert sich dieser Wert leicht. Im Rahmen dieser Validierung nehmen wir den Minimalfall als Gegeben an. Nach unseren Vorüberlegungen sind von 100 Worten also 20 relevant. Der Analysealgorithmus hat bei der Validierung eine Trefferquote von 13,35% für einen normalen Sicherheits-Text und 18,41% für einen BSI-basierten Text. Dies bedeutet, dass der Algorithmus des Riskfinder-Plugins 13 bzw. 18 ’bereinigte’ Treffer findet. ’Bereinigte Treffer’ sind die theoretischen Treffer nach der Trefferquote auf 100 Worte gesehen. Aufgrund dessen können nun die Texte eingestuft werden. Es gibt keine klare Grenze für die Einstufung. In einer empirischen Untersuchung könnte diese Grenzen evaluiert werden. Dies war im Rahmen dieses Artikels aus zeitlichen Gründen nicht möglich. Wie schon oben beschrieben, kann man in normalen Text davon ausgehen, dass 20% der Worte relevant für die Analyse sind. Dieser Wert korrigiert sich leicht nach unten bei einem wissenschaftlichen Text, da in diesem Fall eine komplexe Satzstruktur aufgebaut wird. Die Validierung hat nun gezeigt, dass der Algorithmus des Riskfinder-Plugins ein Ergebnis von 18,41% erzielt, wenn er einen Text untersucht, der auf der Basis des IT-Grundschutz Vokabular beruht. Das ist eine Quote von 92%. Ebenso signifikant ist der Wert von 13,35%, der für einen IT-Sicherheitstext erreicht wurde. Dies ist eine Quote von 66%, es werden zwei von drei sicherheitsrelevanten Wörtern identifiziert. Ein Text, der nicht auf dem Vokabular des IT-Grundschutz basiert, wird erfolgreich als sicherheitsbezogener Text eingestuft.

	Treffer	Anzahl Worte	Treffer pro 100 Worte
Normaler Text	0	512	0
Security Text	11	8237	13,35
Max. Security Text	9	4889	18,41

Tabelle 6.2 Ausgabe der Werte

Fazit

In diesem Artikel wurden Sicherheitsanforderungen am Beispiel von Geschäftsprozessen formuliert und werkzeuggestützt überprüft. Zudem wurde eine Überführung von wichtigen Sicherheitsanforderungen in ein von Werkzeugen zu bearbeitendes Format realisiert. Das Sicherheits-Repository des Riskfinder-Plugins enthält diese Anforderungen in XML-Format. Das Riskfinder-Plugin ist die prototypische Realisierung einer Java-Komponente für das UMLsec-Framework. Für diese Komponente wurde ein Design-Konzept entwickelt und mit einer Dokumentation von Design und Realisation verwirklicht. Dieses Plugin ist in der Lage Spezifikationen einzulesen und auf Sicherheitseigenschaften des UMLsec zu analysieren und zu validieren. Die Implementierung des Riskfinder-Plugins bietet ein

unterstützendes System für ein Risikomanagement und der Verarbeitung von Geschäftsprozessen. Es ermöglicht eine Sicherheitsanalyse auf Basis des IT-Grundschutzes und liefert eine Hilfestellung zur Überarbeitung des Geschäftsprozesses gemäß dem BSI-Grundschutzkataloges. Die Evaluierung zeigt, dass die Ergebnisse, die der Riskfinder-Algorithmus liefert, eine zuverlässige Einstufung der Aktivitäten nach den IT-Grundschutz gibt. Für einen Text aus dem Bereich der IT-Sicherheit, werden 66% der sicherheitsrelevanten Worte identifiziert. Für einen Text der auf dem Vokabular des IT-Grundschutz basiert, werden sogar 92% erkannt. Im Rahmen dieses Artikels wurde ein Werkzeug geschaffen, mit dem es möglich ist, Geschäftsprozesse auf sicherheitsrelevante Aktivitäten zu analysieren und Gefahren zu markieren und Empfehlungen zu geben, die auf Basis des IT-Grundschutzes beruhen. Es werden wenig Kenntnisse des Benutzers vorausgesetzt, aber spezielle Ergebnisse geliefert. Es ist nicht notwendig, wie bei anderen auf dem Markt befindlichen Werkzeugen, große Datenmengen zu erheben, um eine Analyse durchzuführen. Dies ist eine deutliche Erleichterung für den Benutzer.

Ausblick

Als weiterführende Projekte für das Riskfinder-Plugin kann man sich vorstellen, dass man sich mit der Analyse und den Wortvergleichen befasst, sowie mit Erweiterungen, die dem Werkzeug den prototypischen Charakter nehmen. Eine sinnvolle Erweiterung ist eine Implementierung eines Parser für die IT-Grundschutz-Katalog-Inhalte, sodass diese direkt aus den Online-Inhalten aktualisiert werden können. Somit wäre gewährleistet, dass die Sicherheits-Pattern immer auf dem aktuellen Stand sind. Dies würde die manuelle Datenerfassung überflüssig machen und die Möglichkeit bieten, textuelle Inhalte der Gefahren und Maßnahmen des IT-Grundschutz-Katalogs, zu verlinken und dem Benutzer zur Verfügung zu stellen. Für die Erweiterung des Analyse-Algorithmus gibt es mehrere Ansätze. Der erste Ansatz beinhaltet, das Basisvokabular der Sicherheits-Pattern zu erweitern. Zudem kann die Suche nach dem Wortstamm der Worte dazu genutzt werden, um die zu vergleichenden Worte auf einen gemeinsamen Nenner zu bringen. Dies könnte einen erweiterten Trefferraum liefern, durch den eine Zuordnung der Sicherheits-Pattern verbessert werden kann. Die zweite weiterführende Idee ist es, die Treffersuche des Algorithmus zu verändern. Es werden keine Treffer gezählt, sondern die Wortvektoren verglichen und durch eine Kennzahl die Verwandtschaft der Vektoren beschrieben. Somit ist eine Einstufung des Risikos möglich, die sich nicht auf einzelne Worte stützt. Die dritte Idee zur Erweiterung des Algorithmus ist, sich ganz von den Wortvergleichen zu entfernen und ein Augenmerk auf die Struktur zu legen. Es wäre möglich die Struktur eines Diagramms zu analysieren und es dann auf bestimmte Sicherheits-Pattern zu untersuchen, die sich beispielsweise mit dem ’Vier-Augen-Prinzip’ befassen.

Literatur

[ArZy02] Arssenjewa, Zyganowa**.** *Grammatik der deutschen Sprache.* Verlag «Sojuz», 2002.
[BSI10a] BSI. *IT-Grundschutz.* 2010.
[BSI10b] BSI. *Sicherheit durch Verschlüsselung, BSI-Kurzinformationen.* 2010.
[DoWo03] Alberts, Dorofee, Stevens Woody. *Introduction to the Octave Approach*, 2003.
[FeEN09a] Fenz, Ekelhart, Neubauer. *Business Process-Based Resource Importance Determination.* 2009.
[FeEN09b] Ekelhart, Fenz, Neubauer. *Ontologiebasiertes IT Risikomanagement.* 2009.
[Gran02] Gran, Bjorn Axel**.** *Coras, a Plattform for Risk Analys of Security Critical Systems.* 2002.
[Grim03] Grimm, Gebrüder. *Kinder- und Hausmärchen. Gesamtausgabe.* Gondrom Verlag, 2003.
[HöJü08] Sebastian Höhn, Jan Jürjens: Rubacon: automated support for model-based compliance engineering. 30th International Conference on Software Engineering (ICSE 2008), 2008. ACM 2008, S. 875-878

[Hump03] Humpert, Fredrick. *IT-Grundschutz umsetzen mit GSTool. Anleitung und Praxistipps für den erfolgreichen Einsatz des BSI-Standards.* Hanser Verlag, 2003.
[ISO07] ISO 15408**.** *Common Criteria for Information Technology Security Evaluation.* 2007.
[PWXG+07] Petriu,, Woodside, Xu, Georg, France, Bieman, Houmb, Jürjens: Performance analysis of security aspects in UML models. Proceedings of the 6th International Workshop on Software and Performance (WOSP 2007), ACM 2007, S. 91-102
[Jürj05] Jürjens, Jan. *Secure Systems Development with UML.* Springer, 2005
[JüSc08] Jan Jürjens, Jörg Schreck, Peter Bartmann: Model-based security analysis for mobile communications. 30th International Conference on Software Engineering (ICSE 2008). ACM 2008, S. 683-692
[JüSh07] Jan Jürjens, Pasha Shabalin: Tools for secure systems development with UML. International Journal on Software Tools for Technology Transfer (STTT), 9(5-6): 527-544 (2007)
[Knau09] Knaus, Eric. *Sicherheitsrelevante Schlüsselworte.* 2009
[KnSJ09] Eric Knauss, Kurt Schneider, Jan Jürjens**.** *Supporting Requirements Engineers in Recognising Security Issues.* Universität Hannover, 2009.
[MaFl10] Friedmann Mattern, Christian Floerkemeier. *Vom Internet der Computer zum Internet der Dinge.* Institute für Pervasive Computing, 2010.
[UMLs11] UMLsec Werkzeugplattform, http://umlsec.de, 2001-2011