

Anwenden struktureller Nicht-Interferenz zur Sicherheitsanalyse von Workflow-Modellen*

Rafael Accorsi
Albert-Ludwigs-Universität Freiburg, Germany
accorsi@iig.uni-freiburg.de

Abstract: Dieser Beitrag definiert die formalen Grundlagen zur Anwendung von struktureller Nicht-Interferenz für die fundierte Petrinetz-basierte Sicherheitsanalyse von industriellen Workflow-Modellen und damit den ursprünglichen Geschäftsprozessen. Der Ansatz wird anhand eines Beispiels für einen eAuction-Workflow veranschaulicht.

1 Einführung

Betriebswirtschaftliche Geschäftsprozesse werden in IT-gestützten Managementsystemen (BPM) als „Workflows“ abgebildet und ausgeführt. Die Umsetzung dieser Prozesse ermöglicht ihre flexible Anpassung an geschäftliche Veränderungen und die bedarfsgerechte Einbindung externer Ressourcen. Des Weiteren stellt die Nutzung von BPM-Plattformen eine Grundlage für Cloud-Computing dar, welches derzeit als einer der wichtigsten Trends in der Informations- und Kommunikationstechnologie gesehen.

Trotz vorhandenen technischen Know-Hows setzen derzeit nur 23% der deutschen Unternehmen automatisiertes BPM zur Organisation der Wertschöpfungskette mit Zulieferern und Kunden ein [Sta10, S. 27]. Ein wesentlicher Grund hierfür – nicht nur in Deutschland sondern weltweit – stellen Bedenken bezüglich der Datensicherheit dar [AW08, CGJ⁺09]. Wollen Unternehmen ihre Prozesse (partiell oder komplett) auslagern bzw. kollaborativ ausführen, müssen sie über Garantien verfügen, dass verarbeitete Informationen – wie Kunden- und Stammdaten, finanzielle Transaktionen und geschäftliche Interna – geschützt bleiben und die eigenen Prozesse von fremdem Einfluss abgeschottet werden.

InDico ist ein auf Petrinetzen basiertes Rahmenwerk zur automatisierten statischen Analyse von Vertraulichkeitsseigenschaften in Konstellationen nebenläufig ausgeführter Workflows [AW11b] (siehe Abschn. 2). Ähnlich wie bei der Analyse verteilter Sicherheitsprotokolle ist die Annahme, dass bereits zur Designzeit auf der Ebene des Workflow-Modells und ohne Berücksichtigung der technischen Ebene Verletzungen der Sicherheitsseigenschaften entdeckt werden können. Auf dieser Ebene überprüft InDico ein gegebenes Modell und stellt ein entsprechendes Zertifikat aus.

Dieser Beitrag zeigt, dass InDico die Überprüfung von Workflow-Modellen auf verschie-

*Diese Arbeit findet im Rahmen des Schwerpunktprogramms RS3 „Zuverlässig sichere Softwaresysteme“ im Projekt AMBOSS „Automated Business Process Certification Mechanisms“ statt.

dene strukturelle Nicht-Interferenz (NI) [BG09] ermöglicht (Abschn. 4). NI ist eine extensionale Eigenschaft [Ros96], die basierend auf einer Unterteilung des Systems in geheime und öffentliche Teile deren Beziehung bezüglich des Informationsflusses definiert: der „high“ (geheim) Teil eines Systems interferiert *nicht* mit dem „low“ (öffentlich) Teil gwd. die Aktivitäten im „high“ Teilsystem zu keinem beobachtbarem Effekt im „low“ Teilsystem führen [GM82]. Dieser Beitrag zeigt für zwei NI-Eigenschaften, nämlich SNNI und BSNNI [FG01], dass diese als die Abwesenheit von bestimmten „bedrohlichen“ Stellen in einem Petrinetz (oder in InDico, einem IFnet) abgebildet werden können (Abschn. 4.2).

Ein weiterer Beitrag in diesem Artikel ist die Kodierung des Problems, ob ein Netz eine oder mehrere bedrohliche Stellen enthält, als eines der Erreichbarkeit in Petrinetzen (Abschn. 5). Damit wird der Verifikationsansatz von Frau et al. [FGF08] erweitert und dahin verallgemeinert, dass die Verifikation mit bewährten Techniken und Tools für Petrinetze angegangen werden kann. Konkret wird in diesem Beitrag die Erreichbarkeitsmethode von Mayr [May84] basierende auf den „regularly controlled reachability graph“ hingewiesen.

Der Hauptbeitrag dieses Artikels besteht jedoch in der Aussage und dem Nachweis, dass industrielle, nicht-triviale Workflow-Modelle auf NI-Eigenschaften großteils automatisiert überprüft werden können, und dass dies eine mächtige Grundlage für die Zertifizierung von Workflows darstellt. Dies wird anhand eines Beispiels mit einem eAuction-Workflow veranschaulicht (siehe Abschn. 6).

In der Tat stellen NI-Eigenschaften starke, jedoch gerade in diesem Zusammenhang sehr nützliche Isolierungskriterien zwischen den Klassen *high* und *low*. Daher, wenn NI-Eigenschaften nachgewiesen werden können, bedeutet dies, dass die zwei Klassen (auf der Ebene von Workflow-Modellen) voneinander isoliert sind. Andernfalls werden vermeintliche „Informationslecks“ aufgezeigt. Umfassende Untersuchungen im Rahmen von InDico zeigen, dass dieser bezüglich des Informationsflusses als possibilistisch zu bezeichneter Ansatz oft zu „false-positives“ führt, wenn doch Interaktionen zwischen diese Klassen erforderlich sind. Um dieser Problematik entgegenzutreten, besteht die Möglichkeit der Deklassifikation [Sab07], die in InDico zukünftig integriert wird.

2 InDico – Ein Rahmenwerk zur Prozessanalyse

Abb. 1 stellt den InDico-Ansatz schematisch dar (siehe [AW11b] für die Einzelheiten). Dessen Kern bildet IFnet, ein Formalismus zur Beschreibung von Workflows und den darin auftretenden Informationsflüssen. IFnet erweitert die klassische Petrinetz-Definition um Workflow-spezifische Konstrukte wie z.B. Verzweigungsbedingungen, unterscheidbare Objekte und Subjekte. Der Analysevorgang in InDico besteht aus drei aufeinander folgenden Schritten, die in Abb. 1 durch Kästen dargestellt sind:

1. **Transformation nach IFnet.** Im ersten Schritt werden Geschäftsprozesse, die in industriell üblichen Sprachen wie BPEL oder BPMN beschrieben sind, in eine IFnet-Darstellung übersetzt. Gegenwärtig unterstützt InDico die automatische Übersetzung der Kern-Konstrukte sowohl von BPEL als auch von BPMN.
2. **Kodierung von Anforderungen.** Sicherheitsanforderungen müssen für die Infor-

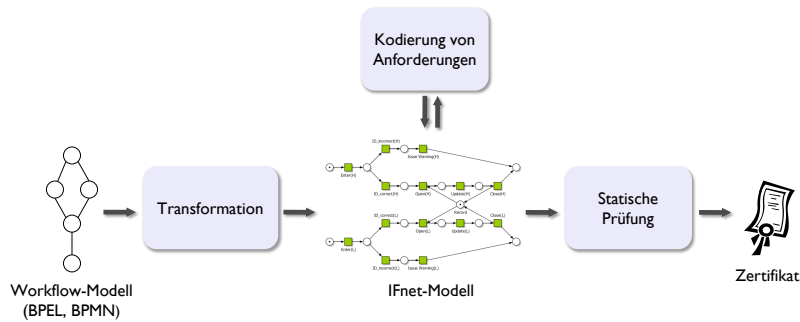


Abbildung 1: Übersicht des InDico-Rahmenwerks.

mationsflussanalyse in ein mehrstufiges Sicherheitsmodell übertragen werden. Dabei werden sowohl die Aktivitäten eines Geschäftsprozesses als auch die Datenobjekte mit Sicherheitsstufen ausgezeichnet, die deren relative Vertraulichkeit bezeichnen und hierarchisch angeordnet sind. Im einfachsten Fall sind dies die zwei Stufen *high* und *low*. InDico bietet verschiedene Strategien, um verbreitete Vertraulichkeitsanforderungen in dieses Stufenmodell zu übertragen.

3. **Statische Prüfung.** Die statische Prüfung stellt fest, ob es in einer Konstellation von IFnet-Modellen zu Informationsflüssen von höher (*High*) eingestuft zu niedriger (*Low*) eingestuft Prozessen bzw. Prozessteilen kommen kann. Dabei wird nicht nur der direkte Austausch von Daten in Betracht gezogen, sondern auch die gegenseitige Beeinflussung von Prozessen, die zur verdeckten Informationsübertragung führen kann. Zentral für deren Erkennung ist das Prinzip der Nicht-Interferenz (NI). InDico implementiert dazu einen Prüfungsalgorithmus, der in einem IFnet zeigen bzw. widerlegen kann, dass es keine Interferenzen zwischen den als *High* und den als *Low* eingestuft Prozessteilen geben kann.

3 IFnet: Information Flow Nets

IFnet basiert auf gefärbten Petrinetzen (CPN), die „klassische“ Petrinetze für die Verarbeitung unterschiedbarer Marken (Tokens) verallgemeinern. Dieser Abschnitt stellt die wesentlichen Aspekte von IFnet vor; siehe [AW11b] für eine detaillierte Definition.

Definition 1 (CPN) Ein CPN ist ein Tupel $N = (P, T, F, C, I, O)$, in dem P eine endliche Menge von Stellen und T eine endliche Menge von Transitionen ist, für die $P \cap T = \emptyset$ gilt. Die Flussrelation F ist eine Menge gerichteter Kanten $F \subseteq (P \times T) \cup (T \times P)$; C steht für die Menge von Farben, die hier durch die natürlichen Zahlen kodiert wird. xFy bedeutet, dass es eine Kante von x nach y gibt. C , I und O werden wie folgt definiert:

1. Die Kapazitätsfunktion $C : P \rightarrow N$ definiert die maximale Anzahl von Marken einer Stelle.
2. Die Eingabefunktion $I : T \times P \times C \rightarrow N$ definiert für jede Transition $t \in T$, jede

- Stelle $i \in P$ mit iFt und Marken-Farbe $c \in N$ die Anzahl der erwarteten Marken.
3. Die Ausgabefunktion $O : T \times P \times C \rightarrow N$ definiert für jede Transition $t \in T$, jede Stelle $o \in P$ mit tFo und Marken-Farbe $c \in N$ die Anzahl produzierter Marken.

Eine Markierung (bzw. ein Zustand) $M : P \times C \rightarrow N$ ist eine Verteilung der Marken auf die Stelle des Netzes. Ein markiertes CPN ist ein Paar (N, M) . \dashv

Transitionen verändern den Zustand des Netzes gemäß der „Flow Relation“ F und der folgenden Schaltregel:

1. Eine Transition $t \in T$ ist in einem Zustand M_1 *aktiviert* (bzw. *schaltbereit*) gwd. jede ihrer Eingangstellen hinreichend viele Marken der jeweils erwarteten Farbe enthält und alle Ausgangstellen eine jeweils ausreichende Kapazität besitzen.
2. Eine aktivierte Transition $t \in T$ kann *schalten* (bzw. *feuern*). Dabei entfernt t sie aus den Eingangsstellen die vorgesehene Menge von Marken und platziert in den Ausgangsstellen die spezifizierten Mengen. Auf diese Weise erreicht das Netz einen Zustand M_2 , der durch die entsprechende Markierung charakterisiert wird.

IFnet erweitern CPN durch spezielle Konstrukte für die Abbildung von Workflows und die Sicherheitsanalyse. Ein IFnet modelliert die Aktivitäten eines Workflows durch Transitionen und Datenobjekte durch Marken. „Schwarze“ Marken stehen für den Kontrollfluss des Workflows, während anders gefärbte Tokens Ressourcen bezeichnen.

Definition 2 (IFnet) Sei $N = (P, T, F, C, C, I, O)$ ein CPN, U eine Subjektmenge, Pr eine Menge von Prädikaten und SC eine endliche Menge von Sicherheitsstufen, die unter der Relation \leq eine Halbordnung bildet. Ein IFnet ist das Tupel $IF = (N, S, A, G, L)$, wobei:

1. die Funktion $S : T \rightarrow U$ ordnet Transitionen Subjekte zu, in derer Verantwortung die Aktivität ausgeführt wird.
2. die Funktion $A : T \times C \rightarrow \{\text{read}, \text{write}\}$ legt fest, ob eine Transition lesend oder schreibend auf ein Eigangsdatum zugreift.
3. Die Funktion $G : T \rightarrow Pr$ ordnet Transitionen Prädikate zu. Beispielsweise bezeichnet $p(\text{red}, \text{green})$ ein Prädikat p , das von Marken der Farben *red* und *green* abhängt. Eine mit einem Prädikat versehene Aktivität kann erst dann schalten, wenn es wahr ist.
4. Die Funktion $L : T \cup C \rightarrow SC$ ordnet Transitionen und gefärbten Marken Sicherheitsstufen zu. In den meisten Fällen gilt $SC = \{\text{low}, \text{high}\}$.

Gegeben sei $x \in P \cup T$. $\bullet x = \{y \mid yFx\}$ bezeichnet das Preset von x , während $x^\bullet = \{y \mid xFy\}$ das Postset. Dies kann auf Mengen von Elementen verallgemeinert werden. \dashv

Eine wesentliche Grundlage für die Analyse des dynamischen Verhaltens eines CPN bzw. IFnet bildet die *Erreichbarkeit* einer Markierung, die gleichzusetzen ist mit der Ausführbarkeit einer bestimmten Sequenz von Transitionen. Eine Markierung M_n ist *erreichbar* von der Anfangsmarkierung M_0 , wenn es eine *Schaltsequenz* (sog. *Trace*) $\sigma = \langle t_1, \dots, t_n \rangle$ gibt, die von M_0 die Markierung M_n produziert. Dies wird mit $M_0[\sigma]M_n$ bezeichnet; für ein Net N , $\mathcal{R}(N, M_0)$ bezeichnet die Menge aller erreichbaren Markierungen ausgehend von M_0 ; die Menge aller möglichen Traces von M_0 wird durch $\mathcal{T}(M_0)$ bezeichnet.

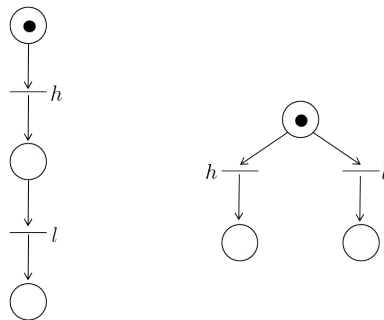


Abbildung 2: Kausal- und Konflikt-Stellen.

Zur Repräsentation von Workflow-Modellen. Um Workflow-Modelle geeignet abzubilden, sind Einschränkungen bezüglich der Form des Modelle. Diese sind im wesentlichen abgeleitet von Workflow-Nets [vdAtH00], beispielsweise, dass das Netz jeweils einen einzigen Anfangs- und Endplatz enthält (sog. „source“ und „sink“ Stellen; siehe [AW11a] für die Einzelheiten). Anhand dieser Bedingungen können Resultate, wie Soundness und Entscheidbarkeit, untersucht und festgelegt werden.

Damit neben manuell spezifizierten auch bestehende Workflow-Modelle mit InDico untersucht werden können, die in einer gebräuchlichen Modellierungssprache wie BPEL oder BPMN beschrieben sind, werden entsprechende Transformationen entwickelt. Aufbauend auf Lohmann [Loh07] und Dijkman et al. [DDO08] bestehen derzeit für BPEL und BPMN automatisierte Transformationen der Kern-Elemente auf IFnet-Muster. Dies umfasst die wesentlichen Konstrukte basierend auf Workflow-Patterns [vdA09], z.B. exclusive Verzweigung und parallele Ausführung.

4 Strukturelle Nicht-Interferenz

Busi and Gorrieri [BG09] führen das Konzept der strukturellen, stellenbasierten Nicht-Interferenz (NI) für elementare (d.h. „pure, simple, safe“ [Mur89]) Petrinetze ein, wobei Frau et al. [FGF08] die Komplexität der Analyse anhand eigens dafür konzipierter Algorithmen festlegen. Dieser Abschnitt gibt einen Überblick über [BG09] (Abschn. 4.1) und überträgt ihre Ergebnisse auf IFnet-Modelle (Abschn. 4.2). Abschnitt 5 reduziert das Problem der Verifikation eines IFnet-Modells auf das der Erreichbarkeit in Petrinetzen.

4.1 Stellenbasierte Nicht-Interferenz

Busi und Gorrieri definieren NI-Eigenschaften anhand der *Abwesenheit* mancher Stellen in einem Petrinetz. Entsprechend die Bezeichnung „Place-based Non-Interference“ (PBNI). Wie Abschn. 5 zeigt, ist es in der Tat möglich, dass solche Stellen zwar im Netz vorhanden

sind, jedoch nie ausgeführt werden. Die Überprüfung eines Netzes besteht daher aus zwei Schritten: der Identifikation potentieller Stellen und, falls welche entdeckt werden, dem Feststellen ihrer Ausführbarkeit. Wenn beide Schritte negativ ausfallen, ist das überprüfte Netz sicher hinsichtlich der in den jeweiligen Stellen kodierten NI-Eigenschaften.

Abb. 2 zeigt die Stellen s , die wesentliche NI-Eigenschaften kodieren. Bei der *kausalen* Stelle, wenn es eine Transition h im Preset von l gibt, die eine Marke in der Stelle s produziert, dann kann der Low-User ableiten, dass h stattgefunden hat (vorausgesetzt, dieser kann l überhaupt beobachten). Es besteht daher eine kausale Abhängigkeit zwischen den Transitionen, deren Abwesenheit die NI-Eigenschaft „Strong Non-Deterministic Non-Interference (SNNI)“ [FG01] zusichert.

Bei einer Konfliktstelle die Stelle s ist im Preset sowohl von h als auch von l , d.h. h und l konkurrieren um die Marke in der Stelle s . Wenn ein Low-User diesen Konflikt kennt und das Schalten der l -Transition beobachten kann, kann er ableiten, dass die h -Transition nicht gefeuert wurde. Daher steht die Stelle s für einen Konflikt zwischen l und h , denn das Schalten von l verhindert das von h und umgekehrt. Die Abwesenheit der Stelle s sichert die NI-Eigenschaft „Bisimulation-based Strong Non-Deterministic Non-Interference (BSNNI)“ [FG01] zu.

Zusammen genommen, die Abwesenheit von Kausal- und Konfliktstellen einem Netz reichen aus, um darin die NI-Eigenschaft „Bisimulation-based Non-Deducibility on Composition (BNDC)“ nachzuweisen [FGF08].

4.2 PBNI in IFnet-Modellen

Um zu garantieren, dass die für elementare Netze entwickelten „Muster“ auch für IFnet und damit auch für die Analyse von Workflow-Modellen verwendet werden können, muss gezeigt werden, dass IFnet-Modelle diese Eigenschaften abbilden können. Dieser Abschnitt stellt das generelle Vorgehen vor und zeigt es für die Kausalstelle, d.h. die SNNI-Eigenschaft. Siehe [AW11b] für die Einzelheiten für die Konfliktstelle.

Das Vorgehen besteht für elementare Netze aus drei Schritten: *erstens* muss das Netz in zwei Partitionen hinsichtlich der Transitionen aufgeteilt werden, die h - und l -Partition; *zweitens* wird das Konzept der Beobachtbarkeit anhand einer Schaltsequenz definiert, d.h. die beobachtbaren Transitionen für einen Low-User; *drittens* werden TCSP-ähnliche Operationen auf Netze definiert, nämlich parallele Komposition und Restriktion in Netzen. Anhand dieser Schritte können anschließend NI-Eigenschaften abgebildet werden.

Die Aufteilung des IFnet-Modells erfolgt durch die Funktion L , welche die Transitionen mit Sicherheitsstufen annotiert. Die Menge T von Transitionen wird unterteilt in $L \subseteq T$ und in $H \subseteq T$ mit $L \cap H = \emptyset$ für die l - und h -Transitionen. Die l -Sicht einer Schaltsequenz definiert, welche Ereignisse low beobachten kann. Darauf aufbauend sind zwei IFnet l -Sicht äquivalent, wenn die produzierten Traces identisch sind für einen Low-User.

Definition 3 (l -Sicht) Gegeben sei ein IFnet-Modell IF . Die l -Sicht Λ_{IF} von einer Schaltsequenz σ von IF ist definiert über die Struktur von σ : $\Lambda_{IF}(\epsilon) = \epsilon$; $\Lambda_{IF}(\sigma t) = \Lambda_{IF}(\sigma)t$

wenn $t \in L$, andernfalls $\Lambda_{IF}(\sigma)$. Diese Definition kann über Mengen von Schaltsequenzen definiert werden. \dashv

Definition 4 (l-Sicht Äquivalenz) Gegeben seien zwei IFnet IF_1 und IF_2 . IF_1 ist l-Sicht äquivalent zu IF_2 , d.h. $IF_1 \stackrel{\Delta}{\approx} IF_2$, gwd. $\Lambda_{IF_1}(\mathcal{T}(IF_1)) = \Lambda_{IF_2}(\mathcal{T}(IF_2))$. \dashv

Parallele Komposition und Restriktion von Netzen sind nicht primitive Operationen für Petrinetze und IFnet. Diese werden wie folgt definiert:

Definition 5 (Parallele Komposition von IFnet) Gegeben seien die IFnet IF_1, IF_2 , bei denen $P_1 \cap P_2 = \emptyset, C_1 \cap C_2 = \emptyset$ und $(L_1 \cup L_2) \cap (H_1 \cup H_2) = \emptyset$. Die parallele Komposition $IF_1 | IF_2$ ist das IFnet $IF_1 | IF_2 = (P_1 \cup P_2, T_1 \cup T_2, F_1 \cup F_2, C_1 \cup C_2, C_1 \cup C_2, I_1 \cup I_2, O_1 \cup O_2, S_1 \cup S_2, A_1 \cup A_2, G_1 \cup G_2, L_1 \cup L_2)$. \dashv

Definition 6 (Restriktion von IFnet) Gegeben sei ein IFnet IF und eine Menge von Transitionen $U \subseteq T$. Die Restriktion von IF bzgl. U ist $IF \setminus U = (P, T', F', C, C, I, O, S, A, G, L)$, so dass $T' = T \setminus U; F' = F \setminus (P \times U \cup U \times P)$. \dashv

Anhand dieser Operationen können die oben genannten NI-Eigenschaften abgebildet werden. Die SNNI-Eigenschaft besagt zum Beispiel, dass ein System sicher ist, wenn die Menge beobachtbarer Transitionen vom Low-User *unabhängig* von den h -Transitionen ist.

Definition 7 (SNNI in IFnet) Gegeben sei IF und $H \subseteq T$. IF ist SNNI-sicher gwd. $IF \stackrel{\Delta}{\approx} IF \setminus H$. \dashv

Ähnlich muss für weitere zu modellierende Eigenschaften definiert werden, welche Menge von Transitionen einem Low-User zugänglich ist. Bei SNNI handelt es sich um eine sehr starke NI-Eigenschaft: aus Sicht eines Low-Users kann nicht unterschieden werden, ob eine h -Transition im System stattfindet. Während eine solche Definition interessant ist, um die Isolierung nebeläufiger Workflows nachzuweisen (daher die parallele Komposition), ist sie aufgrund ihrer Stärke ungeeignet, um die Beziehung zwischen High- und Low-Usern innerhalb eines Workflows zu analysieren. Tatsächlich führt eine Durchsetzung von SNNI in praktischen Beispielen zu mehreren „false-positives“, die anhand eines Deklassifikationsschritts als unbedenklich gekennzeichnet werden können.

5 Verifikation als ein Erreichbarkeitsproblem

Die Verifikation, ob ein gegebenes IFnet (oder Petrinetz) eine der in Abb. 2 aufgeführten Stellen enthält, umfasst einen statischen Schritt zur Entdeckung *potentieller* Stellen und einen dynamischen Schritt zur Feststellung der Ausführbarkeit als potentiell gekennzeichnete Stellen. Tritt dieser Fall ein, rede man von einer *aktiven* Stelle. Nachstehend werden Vorgehen zur Entdeckung von potentiellen und aktiven Stellen vorgestellt.

Der Algorithmus zur Entdeckung potentieller Kausalstellen besteht aus folgenden Schritten (in Anlehnung an [FGF08]):

Definition 8 (Entdeckung potentieller Kausalstellen) Gegeben sei IF und die Prädikate $highPre$, $highPost$ und $lowPost$. Folgende Schritte legen die Menge P^K von potentiellen Kausalstellen in IF fest:

1. für jede $f \in F$:
 - (a) wenn $x \in T$, xFf und $L(x) = high$, dann $highPre(y) = True$ für fFy .
 - (b) andernfalls $lowPost(x) = True$ für xFf .
2. für jede Stelle $p \in P$ von IF ,
wenn $highPre(p) = lowPost(p) = True$, dann füge p in P^K . ↯

Es ist einfach nachzuweisen, dass dieses rein strukturelle Vorgehen zur Entdeckung von potentieller Kausalstellen eine lineare Komplexität $O(p + f)$ aufweist, wobei p für die Anzahl von Stellen und f für Anzahl von Kanten in der Flow-Relation stehen.

Zur Verifikation aktiver Stellen stellen Frau et al. [FGF08] ein Vorgehen basierend auf dem Markierungsgraph des zu untersuchenden Netzes vor. Insbesondere, sie verwenden einen eigens dafür konzipierten, auf Breitensuche basierenden Algorithmus, der für jede Markierung die erreichte Markierung nach dem Schalten aktivierter Transitionen speichert für die Analyse. Die Menge aller solchen Markierung wird dann auf Schaltsequenzen untersucht, in denen eine l -Transition einer h -Transition folgt.

Das Problem der Feststellung, ob eine potentielle Stelle ausführbar (d.h. aktiv) ist, kann jedoch ebenfalls (und auf natürlicher Weise) als das Erreichbarkeitsproblem in Petrinetzen angegangen werden. Das Erreichbarkeitsproblem für ein Netz IF ist die Entscheidung, ob eine Markierung M_n in der Menge der Markierungen enthalten ist, d.h. ob $M_n \in \mathcal{R}(IF, M_0)$. In Anlehnung an Mayr [May84], kann der Markierungsgraph beispielsweise anhand des "klassischen" *Regularly Controlled Reachability Graph* (RCRG)-Algorithmus aufgestellt werden. Dieser Algorithmus verwendet einen endlichen Automaten, um die Anzahl möglicher Schaltsequenzen zu reduzieren.

Durch die Abbildung des Problems in Erreichbarkeit wird die Feststellung der Kausalstellen für die Definition von Markierungen, bei denen die SNNI-Eigenschaft verletzt wird. Im Sinne der RCRG-Algorithmus stellen solche Markierungen sog. „pseudo markings“ dar. Wenn diese erreicht werden können, dann besteht eine Verletzung der SNNI-Eigenschaft.

Definition 9 (Feststellung der Ausführbarkeit einer aktiven Stelle) Gegeben sei IF und die Menge $P^K = \{p_1, \dots, p_n\}$ potentieller Kausalstellen in IF . $M^K = \{M_1, \dots, M_n\}$ ist die aus P^K abgeleitete Menge von entsprechenden Markierungen. Für jede Markierung in M^K , wenn $RCRG(IF, M_i)$ ¹, dann die entsprechende Stelle p_i ist aktiv. Der Trace σ , $M_0[\sigma]M_i$ bildet den Zeugen (bzw. die Evidenz) für diese Verletzung. ↯

Das Erreichbarkeitsproblem ist entscheidbar ist, es weist jedoch eine exponentielle Zeit bezüglich der Anzahl von Stellen [Mur89]. Im Gegensatz zu Frau et al., die für jede Art von

¹Hier bildet der Algorithmus als ein Prädikat; der Digraph A wird der Einfachheit halber weggelassen.

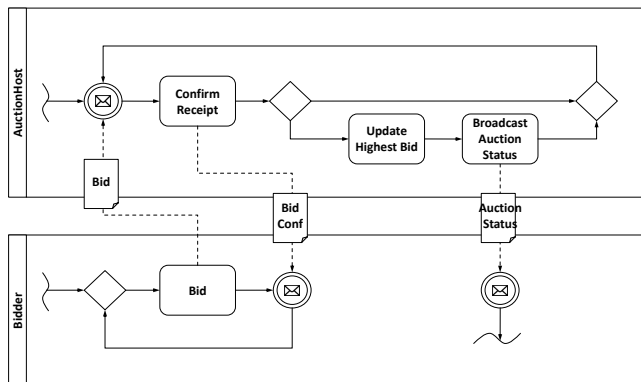


Abbildung 3: Ausschnitt aus einem Auktionsprozess in BPMN-Notation.

Stelle einen neuen Algorithmus verlangen, stellt das obige Vorgehen im Hinblick auf die dynamische Prüfung eine generellere Methode vor, um die Aktivität der Stellen zu überprüfen. Laufende Arbeiten untersuchen hierfür die Nutzung priorisierter Netzen [LP11].

6 Beispiel: Analyse von E-Auktionen

Dieser Abschnitt illustriert den Ansatz und die Anwendung von InDico in einem Beispiel. Abb. 3 zeigt einen Ausschnitt aus zwei interagierenden Prozessen, die eine *Sealed Bid*-Auktion implementieren, bei der die Gebote bis zum Ende der Auktion verschlossen bleiben. Im E-Commerce sind Auktionen ein weit verbreitetes Instrument, um den Preis von Waren und Dienstleistungen festzusetzen, wie z.B. von Konsumgütern und Werbeanzeigen, aber auch von Großhandelswaren wie Elektrizität und Rechenkapazitäten. Um sichere und faire Auktionen zu gewährleisten, ist es entscheidend, die Prozessmodelle bzgl. der Einhaltung entsprechender Richtlinien zu zertifizieren.

Die Prozess-Konfiguration in Abb. 3 beschreibt die Aktivitäten zweier Akteure: in der oberen Hälfte ist der Prozess des Auktionsbetreibers (*Auction Host*) dargestellt, in der unteren Hälfte der eines Bieters (*Bidder*). Das entsprechende IFnet ist in Abb. 4 abgebildet.

Wenn der Bieter ein Gebot abgibt, wird ihm dessen Eingang durch den Betreiber bestätigt und geprüft, ob es das derzeitige Höchstgebot übertrifft. In diesem Fall wird das Höchstgebot erneuert und eine Aktualisierung des Auktionsstatus ausgelöst, über den alle Bieter benachrichtigt werden (z.B. als Email oder Website-Posting).

InDico wird benutzt um die folgenden zwei Sicherheits-Anforderungen zu formalisieren und die gezeigte Prozess-Konfiguration auf ihre Einhaltung zu prüfen:

1. **Isolation von Bieter-Instanzen.** Um eine faire und geheime Auktion zu gewährleisten, darf ein Bieter-Prozess keinerlei Einfluss auf andere Bieter-Prozesse haben.
2. **Vertraulichkeit des Höchstgebotes.** Bieter dürfen vor Ablauf einer Auktion keine

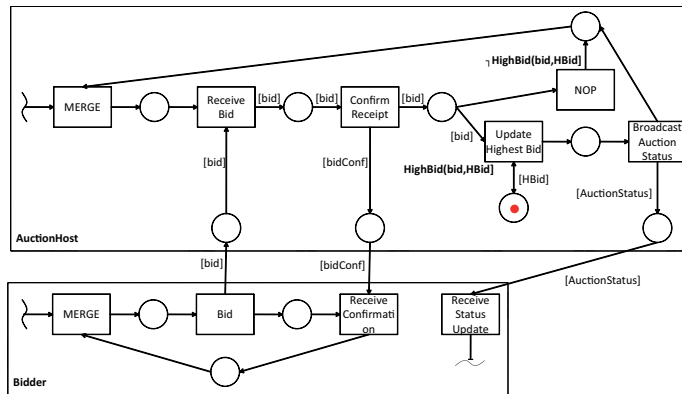


Abbildung 4: IFnet für den Auktionsprozess.

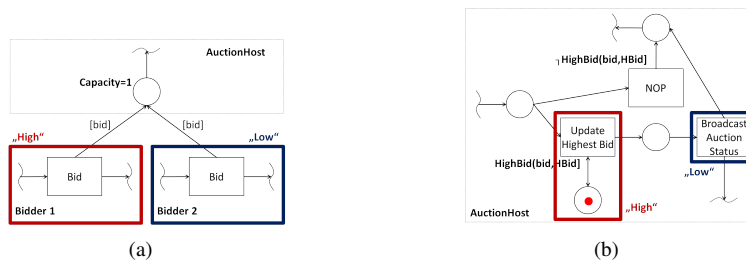


Abbildung 5: Sicherheitsverletzungen im IFnet-Modell der Auktionsprozesse.

Informationen bzgl. des aktuellen Höchstgebotes erlangen.

Wie in Abschnitt 2 beschrieben, werden vor dem eigentlichen Prüfungsvorgang die Prozessmodelle aus Abb. 3 in eine IFnet-Darstellung übersetzt und mit Sicherheitsstufen ausgezeichnet. Abb. 5 zeigt zwei Ausschnitte aus dem entsprechend generierten IFnet-Modell, die im folgenden benutzt werden, um Beispiele für Sicherheitsverletzungen, wie sie von InDico entdeckt werden können, zu illustrieren.

Isolation von Bieter-Instanzen. Abb. 5(a) zeigt den Teil des IFnet Modells, in dem Bieter ihre Gebote an den Auktionsbetreiber übermitteln. Um die Interaktion mehrerer Bieter analysieren zu können, ist der Bieter-Prozess von der entsprechenden Kodierungsstrategie “geklont” (d.h. verdoppelt) worden. Anschließend sind beide Instanzen mit unterschiedlichen Sicherheitsstufen (*high*, *low*) versehen und an der entsprechenden Schnittstelle mit dem Betreiber-Prozess verbunden worden. Die Prüfung des Modells ergibt das folgende Problem, das durch eine Konfliktstelle eintritt: gibt ein Bieter ein Gebot ab, blockiert er die Gebotsabgabe durch den anderen Bieter so lange, bis sein eigenes Gebot verarbeitet worden ist. Diese Interferenz kann sicherheitsrelevant sein, da sie es einem Bieter erlaubt, durch die systematische Abgabe von Geboten Informationen über das Bietverhalten des anderen Bieters zu erhalten (z.B. den Zeitpunkt der Abgabe), oder auch diesen vorübergehend an der Abgabe von Geboten zu hindern. Dies kann unter Umständen – etwa kurz vor Auktionsende – entscheidenden Einfluss auf den Auktionsverlauf haben.

Vertraulichkeit des Höchstgebotes. Abb. 5(b) zeigt den Ausschnitt des IFnet-Modells, in dem die Höhe eines neuen Gebotes geprüft wird. Da das Höchstgebot vertraulich sein soll, ist das entsprechende Datenelement und die Aktivität, die es verarbeitet (*Update Highest Bid*) als *High* markiert worden, während die Aktivität *Broadcast Auction Status* – die für alle Bieter sichtbar ist – die Einstufung *Low* erhalten hat. Ein Problem entsteht aus dem kausalen Zusammenhang zwischen beiden Aktivitäten: aus einer Aktualisierung des Auktionsstatus kann ein Bieter unmittelbar folgern, dass ein neues Höchstgebot eingegangen ist. Durch systematisches Erhöhen der eigenen Gebote bis zu dem Punkt, an dem die Aktualisierung ausgelöst wird, kann er das derzeitige Höchstgebot beliebig genau bestimmen und damit seine Vertraulichkeit brechen.

SWAT Tool-Unterstützung. Die vorgestellte Analyse wurde mit dem SWAT-Tool validiert [AWD11]. SWAT steht für das Open-Source Projekt „Security Workflow Analysis Toolkit“. Es erlaubt die Editierung, Simulation und Analyse von IFnet-Modellen auf Sicherheits- und Compliance-Eigenschaften [ALS11]. Derzeit wird SWAT um die Funktionalität erweitert, auch bestehende Workflows in BPEL und BPMN einzulesen und als IFnet für die Analyse abzubilden. Des Weiteren ist die Einbettung von „plugins“ für die Rekonstruktion von IFnet-Modellen aus entsprechenden Logs vorgesehen, damit Verfahren zu Sicherheits- und Compliance-Audits entworfen und evaluiert werden können.

7 Fazit und Ausblick

Die im Beispiel gezeigten Sicherheitsverletzungen sind Interferenzen, die durch verdeckte Kanäle entstehen (hier: Kontrollfluss). Sie werden durch bisherige Verfahren zur Geschäftsprozessanalyse nicht erkannt und bleiben auch dann bestehen, wenn ein System auf technischer Ebene als sicher zertifiziert worden ist. Die von InDico implementierten Analysen sollen bestehende Verfahren ergänzen und die Ausstellung vollständigerer – und damit zuverlässigerer – Sicherheitszertifikate für Geschäftsprozesse ermöglichen.

Der InDico-Ansatz wird derzeit erweitert, um ein breiteres Spektrum potentieller Sicherheitsverletzungen zu erfassen. Daneben wird an Methoden gearbeitet, um Sicherheitskriterien auf der technischen Ebene (wie die gezeigten Interferenzen) besser mit Sicherheits- und Compliance-Anforderungen auf der organisatorischen Ebene (wie z.B. *Separation of Duty* und *Chinese Wall*) korrelieren zu können.

Literatur

- [ALS11] Rafael Accorsi, Lutz Lowis und Yoshinori Sato. Automatisierte Compliance-Zertifizierung Cloud-basierter Geschäftsprozesse. *Wirtschaftsinformatik*, 53(3):139–149, 2011.
- [AW08] Vijayalakshmi Atluri und Janice Warner. Security for Workflow Systems. In *Handbook of Database Security*, Seiten 213–230. Springer, 2008.

- [AW11a] Rafael Accorsi und Claus Wonnemann. Strong Non-Leak Guarantees for Workflow Models. In *ACM Symposium on Applied Computing*, Seiten 308–314. ACM, 2011.
- [AW11b] Rafael Accorsi und Claus Wonnemann. InDICO: Information Flow Analysis of Business Processes for Confidentiality Requirements. In *ERCIM Workshop on Security and Trust Management*, Jgg. 6710 of *LNCS*, Seiten 194–209. Springer, 2011.
- [AWD11] Rafael Accorsi, Claus Wonnemann und Sebastian Dochow. SWAT: A security workflow toolkit. In *Security Aspects of Process-aware Information Systems*. IEEE, 2011.
- [BG09] Nadia Busi und Roberto Gorrieri. Structural Non-interference in Elementary and Trace Nets. *Mathematical Structures in Computer Science*, 19(6):1065–1090, 2009.
- [CGJ⁺09] Richard Chow, Philippe Golle, Markus Jakobsson, Elaine Shi, Jessica Staddon, Ryusuke Masuoka und Jesus Molina. Controlling data in the cloud: Outsourcing computation without outsourcing control. In *ACM Workshop on Cloud Computing Security*, Seiten 85–90. ACM, 2009.
- [DDO08] Remco M. Dijkman, Marlon Dumas und Chun Ouyang. Semantics and analysis of business process models in BPMN. *Information & Software Technology*, 50(12):1281–1294, 2008.
- [FG01] Riccardo Focardi und Roberto Gorrieri. Classification of security Properties. In *Foundations of Security Analysis and Design*, Jgg. 2171 of *LNCS*, Seiten 331–396. Springer, 2001.
- [FGF08] Simone Frau, Roberto Gorrieri und Carlo Ferigato. Petri Net Security Checker: Structural Non-interference at Work. In *Conference on Formal Aspects in Security and Trust*, Jgg. 5491 of *LNCS*, Seiten 210–225. Springer, 2008.
- [GM82] Joseph Goguen und José Meseguer. Security Policies and Security Models. In *IEEE Symposium on Security and Privacy*, Seiten 11–20, 1982.
- [Loh07] Niels Lohmann. A Feature-Complete Petri Net Semantics for WS-BPEL 2.0. In *Conference on Web Services and Formal Methods*, Jgg. 4937 of *LNCS*, Seiten 77–91. Springer, 2007.
- [LP11] Charles Lakos und Laure Petrucci. Modular State Spaces for Prioritised Petri Nets. In *Monterey Workshop*, Jgg. 6662 of *LNCS*, Seiten 136–156. Springer, 2011.
- [May84] Ernst Mayr. An Algorithm for the General Petri Net Reachability Problem. *SIAM J. Comput.*, 13(3):441–460, 1984.
- [Mur89] Tadao Murata. Petri nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.
- [Ros96] Bill Roscoe. Intensional Specifications of Security Protocols. In *IEEE Computer Security Foundations Workshop*, Seiten 28–38. IEEE, 1996.
- [Sab07] Andrei Sabelfeld. Dimensions of Declassification in Theory and Practice. In *Advances in Computer Science*, Jgg. 4846 of *LNCS*, Seite 1. Springer, 2007.
- [Sta10] Statistisches Bundesamt. *Unternehmen und Arbeitstätten. Nutzung von Informations- und Kommunikationstechnologien in Unternehmen*. Statistisches Bundesamt, 2010.
- [vdA09] Wil van der Aalst. Workflow Patterns. In *Encyclopedia of Database Systems*, Seiten 3557–3558. Springer, 2009.
- [vdAtH00] Wil van der Aalst und Arthur ter Hofstede. Verification Of Workflow Task Structures: A Petri-net-based Approach. *Information Systems*, 25(1):43–69, 2000.