

# Informationsflusskontrolle als Grundlage für die Sicherheit von Multi-Agenten-Systemen

Dieter Hutter, Heiko Mantel und Axel Schairer

Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI GmbH),  
Stuhlsatzenhausweg 3, 66123 Saarbrücken  
{hutter|mantel|schairer}@dfki.de

## Zusammenfassung

Zur Lösung komplexer verteilter Probleme werden vermehrt Agenten-Systeme eingesetzt, bei denen Teile der Funktionalität typischerweise dezentral über Verhandlungen zwischen einzelnen Agenten realisiert werden. Beim Nachweis der Sicherheit solcher Systeme werden somit Eigenschaften der Verhandlungsführung oder allgemein der Kommunikation zwischen Agenten sicherheitskritisch. In diesem Artikel wird exemplarisch für den Bereich des Comparison-Shoppings aufgezeigt, wie man ausgehend von informell spezifizierten Sicherheitsanforderungen an ein Agenten-System schrittweise zu formal verifizierbaren Bedingungen im Sinne eines Informationsflussansatzes gelangt.

## 1. Einleitung

In den letzten Jahren werden Multi-Agenten-Systeme vermehrt zur Lösung von verteilten Problemen eingesetzt. Im Gegensatz zu herkömmlichen Verfahren, bei denen eine zentrale Instanz die Verwaltung von kritischen oder beschränkten Ressourcen vornimmt, erfolgt die Konfliktlösung bei Multi-Agenten-Systemen auf Basis von Verhandlungen zwischen den betroffenen Agenten. Die konkurrierenden Interessen der im System modellierten Beteiligten spiegeln sich auf natürliche Weise in den einzelnen Agenten des Systems wider.

Durch die Konkurrenz der beteiligten Agenten entstehen entsprechende Sicherheitsbedürfnisse der Beteiligten hinsichtlich des Informationsaustausches zwischen Agenten untereinander aber auch zwischen Agenten und Agenten-Plattform. Sicherheitsanforderungen an das Gesamtsystem reduzieren sich zum einen auf Sicherheitsanforderungen an die einzelnen Agenten und zum anderen auf Sicherheitsanforderungen an den Verhandlungsablauf bzw. an die Kommunikation zwischen den Agenten.

Dieser Artikel beschäftigt sich mit dem Prozess, aus informell gegebenen Sicherheitseigenschaften für ein Multi-Agenten-System schrittweise formale Anforderungen an die Teilkomponenten des Systems herzuleiten. Als Beispiel verwenden wir ein einfaches Comparison-Shopping Szenario [2,15], in dem Käufer-Agenten verschiedene Händler-Agenten kontaktieren, um deren Angebot für ein gewünschtes Produkt zu erhalten. Ausgehend von den informell gehaltenen Sicherheitsinteressen eines Händlers werden wir Sicherheitsanforderungen an die beteiligten Agenten bestimmen, die dann ihrerseits als Grundlage zur formalen Spezifikation von Sicherheitseigenschaften dienen werden. Eine formale Spezifikation der Sicherheitseigenschaften schließt zum einen mögliche Mehrdeutigkeiten der Anforderungsspezifikation aus und zum anderen erlaubt sie es, die geforderten Eigenschaften zu verifizieren, d.h. mit mathematischen Mitteln nachzuweisen, dass ein System diese Eigenschaften besitzt.

Wir werden in diesem Artikel die Frage diskutieren, wie Sicherheitsanforderungen adäquat in einem formalen Rahmen repräsentiert werden können und werden zur formalen Spezifikation der Sicherheitseigenschaften aus folgenden Gründen einen Informationsflussansatz verwenden:

- Informationsflusspolitiken eignen sich für eine elegante und rigorose Formalisierung von Sicherheitsanforderung (insbesondere von Anforderungen an die Vertraulichkeit)
- Die Theorie der Informationsflusseigenschaften ist für unsere Zwecke hinreichend gut erforscht und auch für die Verwendung in der Praxis geeignet. Insbesondere bietet sie einen Rahmen für ein Herunterbrechen der globalen Sicherheitsanforderungen eines Multi-Agenten-Systems auf lokale Anforderungen an die beteiligten Komponenten.

- Der Informationsflussansatz ist flexibel genug, um auch komplexere Szenarios (als das hier betrachtete) zu analysieren. Ein Beispiel wäre hier die explizite Modellierung der Agenten-Plattform bzw. die Kommunikation zwischen den Agenten über die Plattform.

Wir beginnen im Abschnitt 2 mit einer informellen Darstellung des betrachteten Comparison-Shopping Szenarios und einer Beschreibung der darin auftretenden Sicherheitsproblematiken. In Abschnitt 3 leiten wir die sich daraus ergebenden Sicherheitsanforderungen an das System her und formulieren diese möglichst präzise als Vertraulichkeitseigenschaften. Die Vertraulichkeitseigenschaften werden dann in Abschnitt 4 als Informationsflusseigenschaften eines Event-Systems formal erfasst. Die daraus für mögliche Implementierungen resultierenden Ergebnisse und Schlussfolgerungen diskutieren wir in Abschnitt 5.

## 2. Comparison Shopping

Durch die große Anzahl von Warenangeboten im Internet und deren unübersichtliche Darstellung ist es heute für einen Käufer kompliziert geworden, aus der Vielzahl der Angebote ein für ihn passendes auszuwählen. Insbesondere ist es praktisch fast unmöglich, das jeweils günstigste, aktuell vorhandene Angebot herauszusuchen. Hier ist eine automatische, d.h. informatische Lösung hilfreich, wie sie z.B. durch das *Comparison Shopping* [2, 15] realisiert wird. Dabei werden potentielle Käufer durch Agenten repräsentiert, die anhand vorgegebener Kriterien das Suchen und Vergleichen von Angeboten übernehmen. Sogenannte *Käufer-Agenten* begeben sich auf virtuelle Marktplätze, an denen verschiedene *Händler-Agenten* ihre Angebote feilhalten. Der *Matchmaker* übernimmt die Rolle eines Maklers, der potentielle Käufer anhand ihrer Kriterien an geeignete Händler-Agenten weitervermittelt.

Wird ein Käufer-Agent zum Beispiel zu fünf unterschiedlichen Händlern vermittelt, so kontaktiert er deren Agenten, um entsprechende Angebote zu erhalten. Aus den erhaltenen Angeboten wählt er anhand seiner Bedürfnisse und Kaufkriterien das Geeignetste aus. Die Kaufkriterien werden vom Besitzer des Käufer-Agenten im Voraus spezifiziert und dem Käufer-Agenten mitgegeben. Mögliche Kriterien sind beispielsweise der Preis, die Garantiebedingungen, oder etwa der Funktionsumfang des Produktes. Um die Käuferpräferenzen zu spezifizieren, eignen sich z.B. Regelmengen oder Bewertungsfunktionen.

Händler und Käufer haben in der Regel unterschiedliche Interessen, die sich nicht nur auf den Umfang und die Qualität des Angebots beschränken. Während der Händler daran interessiert ist, ein Profil seiner Kundschaft aufzubauen, um langfristig sein Angebot auf die Bedürfnisse seiner Kundschaft zu orientieren, wollen in der Regel potentielle Käufer (zunächst) anonym auftreten, um ihre Privatsphäre nicht preisgeben zu müssen. Verschiedene Käufer-Agenten können in Konkurrenz um ein beschränktes Angebot eines Händlers treten und umgekehrt werden Händler versucht sein, durch ein attraktives Angebot ihre Mitbewerber zu unterbieten. Das Wissen um das Angebot eines Konkurrenten bedeutet hier einen massiven strategischen Vorteil. Konkurrierende Händler werden sich also nur dann auf einer gemeinsamen Plattform bewegen, wenn die Wahrung solcher strategischen Geheimnisse gewährleistet ist. Da sowohl der Handel als auch die Datensammlung und –übertragung auf der Plattform elektronisch erfolgen, ist das Nutzen des strategischen Vorteils sehr viel schneller und effizienter möglich als im herkömmlichen nicht-elektronischen Handel. Daher ist die Einhaltung dieser Anforderung in unserem Szenario sehr viel wichtiger. Andererseits sammeln Kunden Angebote und könnten im Laufe ihrer Verhandlungen auch die Angebote konkurrierender Händler zur Durchsetzung ihrer Interessen einsetzen, was zur Folge hätte, dass einem Händler konkrete Konkurrenzangebote zur Kenntnis kommen. Aber auch aus der Sicht eines Käufers ist die Liste der Angebote, die er von den Händlern erhalten hat, schützenswert. Schließlich haben sowohl Händler als auch Käufer ein vitales Interesse an der Integrität des Matchmakers, der potentielle Käufer an geeignete Händler vermitteln soll. Der Matchmaker sollte Anfragen der Käufer und Änderungswünsche der Händler korrekt bearbeitet, d.h. keine Anfragen unterdrücken oder wissentlich falsch beantworten. Auch darf der Matchmaker seine Zuordnung von Waren zu einem Händler nur aufgrund entsprechender Nachrichten des betroffenen Händlers ändern.

Für eine weitergehende Analyse der Sicherheitsanforderungen ist es zunächst wichtig, dass wir näher auf die Realisierung des betrachteten Multi-Agenten-Systems eingehen.

Wir gehen dabei von einer Agenten-Plattform aus, auf der sich Matchmaker, Händler und Käufer gemeinsam aufhalten. Eine solche *Virtual Mall* bietet eine Schnittstelle zum Internet, über die potentielle Händler oder Käufer zur Verfügung stehende, generische Händler- oder Käufer-Agenten mit ihrer persönlichen Angebotspalette oder ihren Kaufwünschen instantiiieren und auf der Plattform zur Ausfüh-

rung bringen können. Der unabhängige Betreiber der Plattform hat für einen reibungslosen und sicheren Betrieb der Plattform zu sorgen.

Die einzelnen Agenten-Arten, Käufer  $A$ , Händler  $H$  und Matchmaker  $M$ , könnten sich dabei wie folgt verhalten: Um Angebote einzuholen, schickt ein Käufer-Agent  $A$  zunächst eine Beschreibung  $R$  seines Kaufwunsches an den Matchmaker  $M$ . Der Matchmaker antwortet hierauf mit einer Liste von potentiell geeigneten Händlern  $H, H', H''$ , die (wahrscheinlich) das gewünschte Produkt zu den spezifizierten Konditionen anbieten können. Der Käufer-Agent erstellt sich daraus eine Tabelle, in der die vom Matchmaker genannten Händler  $H, H', H''$  mit den zugehörigen Angeboten  $O_H, O_{H'}, O_{H''}$  erfasst werden. Für die Erstellung der Tabelleneinträge sendet der Käufer-Agent an die vom Matchmaker genannten Händler-Agenten Anforderungen, ein Angebot für den Kaufwunsch abzugeben. Auf die Anfrage antwortet ein Händler  $H$  mit einem Angebot  $O_H$ , das der Käufer-Agent nach Erhalt in seine Tabelle einträgt. Der Käufer-Agent verfügt über eine Bewertungsfunktion, die aufgrund der Tabelleneinträge festlegt, ob bereits ein zufriedenstellendes Angebot abgegeben wurde und damit die Angebotseinholung beendet werden kann. Wurde kein zufriedenstellendes Angebot gefunden, werden weitere bisher nicht befragte Händler-Agenten konsultiert. Sind alle Angebote eingeholt worden und trotzdem kein akzeptables Angebot gefunden, kann der Käufer-Agent einen modifizierten Kaufwunsch  $R'$  an die Händler-Agenten schicken, in dem beispielsweise die Konditionen, zu der das Produkt gekauft werden soll, innerhalb vorgegebener Grenzen modifiziert wurden.

Ein Händler-Agent  $H$  sendet initial dem Matchmaker seine Produktpalette und wartet dann auf Anfragen von Käufer-Agenten. Er besitzt intern eine Tabelle der von ihm lieferbaren Waren mit den entsprechenden Konditionen. Auf Anfrage  $R$  eines Käufer-Agenten sendet der Händler-Agent sein Angebot  $O_H$  zurück. Der Händler-Agent verfügt über eine Historie, in der abgespeichert ist, welche Titel zu welchen Konditionen welchen Käufer-Agenten angeboten bzw. verkauft wurden. In Abhängigkeit von der Historie und dem vom Käufer-Agenten zugesandtem Angebot berechnet der Händler-Agent bei einer wiederholten Anfrage sein neues Angebot und sendet dem Käufer-Agenten das überarbeitete Angebot zu. Der Matchmaker  $M$  beantwortet einerseits Anfragen von Käufer-Agenten anhand der momentan bei ihm gespeicherten Liste der Händler und ihrer aktuellen Produktpaletten. Andererseits bearbeitet er Nachrichten von Händler-Agenten, in denen sie ihre aktuelle Produktpalette dem Matchmaker mitteilen. Weiterhin beantwortet der Matchmaker Anfragen von Käufer-Agenten anhand der momentan bei ihm gespeicherten Liste der Händler und ihrer aktuellen Produktpaletten.

### 3. Formulierung der Sicherheitsanforderungen

Für den Betreiber einer Comparison-Shopping Plattform ist es natürlich von strategischem Interesse, das Vertrauen von potentiellen Händlern und Käufern für seine Plattform zu gewinnen und sie von der Sicherheit der Plattform zu überzeugen. Dabei hat er zu beachten, dass Händler-Agenten, Käufer-Agenten und Matchmaker oft konkurrierende Interessen haben. Insbesondere können Sicherheitseigenschaften eines Agenten mit Interessen anderer Agenten in Konflikt stehen. Im Folgenden möchten wir diese Sicherheitsinteressen genauer beleuchten.<sup>1</sup>

Ein wesentliches Sicherheitsinteresse des Matchmakers besteht darin, dass er von Händler-Agenten korrekte Informationen über die angebotene Produktpalette bekommt. Händler könnten sonst mit falschen Angaben Kunden anlocken und durch Verhandlungen versuchen, Käufer-Agenten zu Modifikationen ihrer Kaufwünsche zu "überreden". Das könnte z.B. dazu führen, dass Kunden, die mit diesem Vorgehen unzufrieden sind, zukünftig lieber eine konkurrierende Virtual Mall aufsuchen, in der dieses Problem nicht auftritt.

Der Käufer-Agent hat ein vitales Interesse an der Integrität des Angebots, d.h. dass der Händler-Agent bei einem Vertragsabschluss das Produkt zu den angebotenen Konditionen auch wirklich verkauft. Wegen der Konkurrenzsituation zwischen interessierten Käufer-Agenten (speziell bei einem begrenzten Angebot), möchte der Käufer-Agent seine Tabelle der Angebote außerdem gegenüber anderen Käufer-Agenten geheim halten. Zusätzlich ist der Käufer-Agent abhängig von der Integrität des Matchmakers, da der Matchmaker letztlich die für den Kauf eines Produktes in Frage kommenden Händler bestimmt.

Der Händler-Agent hat ebenfalls ein vitales Interesse an der Integrität des Matchmakers, da er sonst potentielle Kunden verlieren könnte. Weiterhin will der Händler-Agent nicht, dass seine Angebote ande-

---

<sup>1</sup>Für eine eingehende Diskussion der Sicherheitsinteressen von Agenten beim Comparison Shopping sei der Leser auf [13] verwiesen.

ren bekannt werden, außer den Kunden, denen er sie explizit gegeben hat. Dieses umfasst andere Verkäufer und Kunden, denen er dieses Angebot nicht explizit gegeben hat, sowie deren Agenten.

Ausgehend von dem im vorherigen Abschnitt beschriebenen Szenario wollen wir nun aus den Sicherheitsinteressen beispielhaft zentrale Sicherheitsanforderungen in Termini der beteiligten Agenten-Arten spezifizieren. Dazu werden wir zuerst die Sicherheitsanforderungen informell herleiten und dann schrittweise versuchen, diese präziser zu beschreiben.

### 3.1 Herleitung der Sicherheitsanforderungen

Betrachtet man die Händlerseite, so wollen, wie zuvor bereits erläutert, Händler nicht, dass die von ihnen gemachten Angebote ihren Konkurrenten bekannt werden, da diese Informationen für den Konkurrenten einen strategischen Vorteil bedeuten würden. Daraus ergibt sich zum einen die Sicherheitsanforderung an Händler-Agenten, dass sie Angebote nur an anfragende Kunden schicken dürfen. Zum anderen ergibt sich aber auch eine Sicherheitsanforderung an die Käufer-Agenten, nämlich dass diese keine Informationen über die von einem Händler-Agenten erhaltenen Angebote an andere Agenten weitergeben dürfen.

Die Anforderungen an das globale Szenario, nämlich dass Angebote eines Händlers nur den Agenten bekannt werden, denen der Händler die Angebote durch seinen Agenten geschickt hat (und auch schicken wollte), ist somit auf lokale Anforderungen an den Agenten des Händlers und die Agenten der Käufer heruntergebrochen worden. Dieses Beispiel zeigt auch, dass Sicherheitsinteressen eines Teilnehmers, hier des Händlers, nicht nur zu Sicherheitsanforderungen an seinen eigenen Agenten, sondern auch zu Sicherheitsanforderungen an Agenten anderer Teilnehmer, hier an die Agenten der Käufer, führen können.

In ähnlicher Weise können auch andere globale Sicherheitsanforderungen auf lokale Sicherheitsanforderungen an die *einzelnen beteiligten* Agenten reduziert werden. Die globale Sicherheitsanforderung eines Käufer-Agenten  $K$ , dass seine Tabelle der Angebote anderen Agenten nicht bekannt wird, resultiert zum Beispiel in den lokalen Sicherheitsanforderungen an einen Händler-Agenten  $H$ , der  $K$  ein Angebot  $O_H$  gemacht hat, dass er  $O_H$  keinem anderen Agenten mitteilen darf.

Im weiteren werden wir uns auf Sicherheitsanforderungen konzentrieren, die sich aus den Sicherheitsinteressen des Händlers ergeben. Unter anderem ergeben sich folgende lokale Sicherheitsanforderungen an die einzelnen Agenten. Im Folgenden lassen wir vorerst die präzise Bedeutung von „vertraulich“ offen und werden erst im nachfolgenden Abschnitt auf diesen Punkt näher eingehen.

#### Lokale Sicherheitsanforderungen an den Käufer-Agenten.

- Die Angebote, die ein *Käufer-Agent* von einem Händler oder seinem Agenten bekommen hat, muss der *Käufer-Agent* gegenüber anderen Händlern und ihren Agenten „vertraulich“ halten.
- Die Angebote, die ein *Käufer-Agent* von einem Händler oder seinem Agenten bekommen hat, muss der *Käufer-Agent* gegenüber anderen Käufern und ihren Agenten „vertraulich“ halten.

#### Lokale Sicherheitsanforderungen an den Händler-Agenten.

- Angebote, die ein *Händler-Agent* einem Käufer-Agenten gemacht hat, muss der *Händler-Agent* gegenüber anderen Händlern und ihren Agenten „vertraulich“ halten.
- Angebote, die ein *Händler-Agent* einem Käufer-Agenten gemacht hat, muss der *Händler-Agent* gegenüber anderen Käufern und ihren Agenten „vertraulich“ halten.

Diese Liste von Sicherheitsanforderungen ist nicht vollständig, sondern als beispielhafte Illustration solcher Anforderungen zu verstehen. Eine weitere Sicherheitsanforderung die sich aus dem Sicherheitsinteresse eines Händlers  $H$  ergibt, ist zum Beispiel, dass der Matchmaker die Zuordnung von Waren zu  $H$  nur auf eine entsprechende Anfrage von  $H$  ändert und dass solche Änderungen in korrekter Weise durchgeführt werden. Eigenschaften von dieser Form lassen sich mit Safety-Eigenschaften [1] modellieren. Wie man Safety-Eigenschaften modelliert ist wohlbekannt und soll hier nicht weiter ausgeführt werden. Unser Fokus in diesem Artikel liegt auf der (konzeptionell schwierigeren) Formalisierung von Vertraulichkeitsanforderungen.

### 3.2 Präzisierung der Vertraulichkeitseigenschaften

Im vorhergehenden Abschnitt haben wir die Sicherheitsanforderungen an die einzelnen Agenten beim Comparison Shopping präzise gefasst. Dabei haben wir uns allerdings einen Freiheitsgrad gelassen, indem wir den Begriff „Vertraulichkeit“ an verschiedenen Stellen verwendet haben, ohne näher zu spezifizieren, was genau damit gemeint ist. Da es verschiedene in Frage kommende Interpretationen von „Vertraulichkeit“ gibt, müssen wir jetzt konkret angeben, welche Interpretation diese intuitiv gefasste Eigenschaft möglichst genau trifft. Um aufzuzeigen, was bei der Interpretation von „Vertraulichkeit“ wichtig ist, betrachten wir zuerst eine naheliegende Interpretation, die – wie wir später aufzeigen werden – einige Probleme in sich birgt.

Als Beispiel benutzen wir folgende Sicherheitsanforderung an den Käufer-Agenten aus dem vorigen Abschnitt:

Die Angebote, die ein Käufer-Agent von einem Händler oder seinem Agenten bekommen hat, muss der Käufer-Agent gegenüber anderen Händlern und ihren Agenten „vertraulich“ halten.

Vertraulichkeit der Angebote könnte hier interpretiert werden als die Anforderung:

Ein Käufer-Agent darf Angebote, die er von einem Händler (bzw. dessen Agenten) erhalten hat, nicht an andere Händler (bzw. deren Agenten) *verschicken*. Das heißt, es dürfen keine Nachrichten an andere Händler versandt werden, die diese Angebote enthalten.

Leider ist diese naheliegende Interpretation in unserem Kontext nicht gut geeignet. Dafür gibt es mehrere Gründe. Ein Agent könnte ein erhaltenes Angebot modifizieren, z.B. indem er es mit dem öffentlichen Schlüssel eines anderen Agenten verschlüsselt, und es erst danach an diesen Agenten verschickt. Obwohl das offensichtlich eine Verletzung der Vertraulichkeit bedeutet (der Empfänger der Nachricht kann das Angebot durch Entschlüsseln rekonstruieren), liegt unter der zuvor angegebenen Interpretation von „Vertraulichkeit“ keine Verletzung der Sicherheitseigenschaft vor. Dieses Beispiel zeigt, dass die oben angegebene Interpretation von „Vertraulichkeit“ zu schwach ist.

Im Allgemeinen lässt sich anhand des Textes einer Nachricht nicht entscheiden, ob sie eine Verschlüsselung des geheimen Angebotes ist. Vielmehr ist es entscheidend, dass der Agent das Geheimnis nicht in die Nachricht einfließen lässt. Die Angebote, die ein Käufer-Agent von einem Händler *H* erhalten hat, dürfen also nicht beeinflussen, welche Nachrichten an andere Händler geschickt werden. In anderen Worten:

Die Nachrichten, die ein Käufer-Agent an Händler (außer an *H*) verschickt, dürfen nicht von Angeboten *abhängen*, die er von *H* erhalten hat.

Das heißt, Vertraulichkeit ist eine Abhängigkeitseigenschaft. In *ad hoc*-Formulierungen von Vertraulichkeit, wie z. B. in der zuerst angegebenen, wird diese Tatsache leider leicht vergessen.

## 4. Vertraulichkeit als Abhängigkeitseigenschaft

### 4.1 Sicherheitsanforderungen als Abhängigkeitseigenschaften

Für das Beispiel, dass die von einem Käufer-Agenten gespeicherten Angebote anderen Händlern gegenüber vertraulich sein sollen, stellt sich also die Frage: *Kann ein Händler durch Nachrichten, die er von einem Käufer bekommt, schließen, dass dieser schon ein (bestimmtes) anderes Angebot bekommen hat?* Das ist dann der Fall, wenn eine Abhängigkeit zwischen dem vorher eingegangenen Angebot und den an den Händler versandten Nachrichten besteht. Oder: *Wären die Beobachtungen des Händlers auch möglich gewesen, wenn der Käufer das fragliche Angebot gar nicht erst bekommen hätte?* In diesem Falle würden die an den Händler geschickten Nachrichten nicht die Vertraulichkeitsanforderung verletzen.

Für das konkrete Beispiel „Angebot vertraulich gegenüber anderem Händler“ ist also die zu schützende *vertrauliche Information* das Eingehen eines Angebotes beim Käufer. Wenn das Eingehen des Angebotes geheim bleibt, ist insbesondere sichergestellt, dass auch der Inhalt des Angebotes geheim bleibt. Die *Beobachtungen*, aufgrund derer es nicht möglich sein darf, die vertrauliche Information zu erschließen, sind die Nachrichten, die der Käufer an andere Händler schickt.

Wie aus diesem Beispiel hervorgeht, ist es also notwendig, für alle Vertraulichkeitsanforderungen präzise zu bestimmen, was die Informationen sind, die einem Beobachter gegenüber vertraulich sein sollen, und welche Beobachtungsmöglichkeiten der identifizierte Beobachter hat, um auf die vertrauliche

Information zu schließen. In Termini der einzelnen Nachrichten zwischen Agenten lassen sich die folgenden Formulierungen von lokalen Sicherheitsanforderungen für Käufer-Agenten und Händler-Agenten ableiten:

**Präzisierte Sicherheitsanforderungen an den Käufer-Agenten.**

- Die Nachrichten, die ein Käufer-Agent an Händler (außer an  $H$ ) verschickt, dürfen nicht von Angeboten abhängen, die er von Händler  $H$  erhalten hat.
- Die Nachrichten, die ein Käufer-Agent an andere Käufer verschickt, dürfen nicht von Angeboten abhängen, die er von Händlern erhalten hat.

**Präzisierte Sicherheitsanforderungen an den Händler-Agenten.**

- Die Nachrichten, die ein Händler-Agent an andere Händler verschickt, dürfen nicht von Angeboten abhängen, die er abgegeben hat.
- Die Nachrichten, die ein Händler-Agent an einen Käufer  $K$  verschickt, dürfen nicht von Angeboten abhängen, die er einem anderen Käufer  $K'$  gesendet hat.

Im Vergleich zu den in Abschnitt 3.1 angegebenen Formulierungen dieser Sicherheitsanforderungen, sind die hier angegebenen präziser, da sie genau angeben, was mit „vertraulich“ gemeint ist.

**4.2 Allgemeiner Ansatz**

Die im obigen Absatz angegebenen Formulierungen der Sicherheitsanforderungen als Abhängigkeits-eigenschaften haben gemeinsam, dass bestimmte Beobachtungen nicht von einem bestimmten Geheimnis abhängig sein dürfen. Welches diese Beobachtungen sind und was die Geheimnisse sind, unterscheidet sich zwischen den verschiedenen Sicherheitsanforderungen. Trotz dieser Unterschiede, folgen alle diese Formulierungen demselben Schema:

*Die Beobachtung von ... darf nicht abhängen von ...*

Um diese Aussage formalisieren zu können, ist es notwendig, genauer zu beschreiben, was wir unter den Beobachtungen von Agenten verstehen. Während des Ablaufs eines Agenten-Programms nimmt ein Agent alle Ereignisse wahr, die seine Schnittstelle betreffen<sup>2</sup>. Ein Händler-Agent  $H$  könnte z. B. wahrnehmen, dass er zuerst eine Nachricht  $R_1$  vom Käufer-Agenten  $A$  erhält, danach eine Nachricht  $O_2$  an  $A$  zurückschickt, und schließlich eine weitere Nachricht  $R_3$  von einem anderen Käufer-Agenten  $B$  empfängt.<sup>3</sup> Wir beschreiben also die Beobachtungen eines Agenten während eines Ablaufes durch die Abfolge von allen Ereignissen, die an seiner Schnittstelle geschehen. Formal lässt sich die Beobachtung eines Agenten als eine Sequenz von Termen beschreiben, wie z. B.

$$\langle receive_H(A, R_1), send_H(A, O_2), receive_H(B, R_3) \rangle. \tag{1}$$

Jeder Term in der Sequenz beschreibt ein atomares Ereignis, d.h. einen sogenannten *Event*.

Das angegebene Beispiel (1) beschreibt die Ereignisse an der Schnittstelle von  $H$ : Ein Event  $receive_H(A, R_1)$  steht für das Ereignis, dass die Nachricht  $R_1$  mit Absender  $A$  bei  $H$  ankommt. Entsprechend kann  $A$  an seiner Schnittstelle einen Event  $send_A(H, R_1)$  beobachten, der für das Ereignis steht, dass  $A$  die Nachricht an  $H$  abschickt. Diese Ereignisse werden bei der Modellierung geeignet miteinander verknüpft, so dass das gewünschte Kommunikationsverhalten modelliert wird.<sup>4</sup>

Betrachtet man eine Menge von Agenten, etwa  $\{H', H''\}$ , deren Programme simultan ablaufen, so lässt sich die Beobachtung jedes einzelnen dieser Agenten auf die oben angegebene Art und Weise durch eine Sequenz von Events angeben. Gemeinschaftliche Beobachtungen von Agentengruppen lassen sich ebenfalls als Sequenzen von Events angeben, genauer als Sequenzen von den Events, die an der Schnittstelle irgendeines Agenten aus der Gruppe geschehen. Diese Sichtweise ist etwa nützlich, wenn betrachtet wird, welche Beobachtungen prinzipiell von einer Koalition der Händler-Agenten  $H'$  und  $H''$  gemacht werden können. Es könnte sich z. B.

<sup>2</sup>Natürlich nimmt ein Agent auch die Ereignisse wahr, die er intern ausführt. Das ist im Moment aber nicht relevant.

<sup>3</sup> Im folgenden gehen wir implizit davon aus, dass Nachrichten authentisch sind. D.h., zum Beispiel, ein Ereignis  $receive_H(A, R_1)$  bedeutet, dass  $H$  die Nachricht  $R_1$  tatsächlich von  $A$  erhalten hat.

<sup>4</sup>Siehe auch Fussnote 4.

$$\langle receive_H(A, R_1), receive_{H'}(A, R_1), send_H(A, O_2), receive_{H'}(B, R_3) \rangle \quad (2)$$

ergeben, falls  $H'$  und  $H''$  nacheinander jeweils die Anfrage  $R_1$  von  $A$  empfangen,  $H'$  diese Anfrage beantwortet, und schließlich  $H''$  nochmals eine Anfrage von einem anderen Käufer-Agenten  $B$  empfängt. Man beachte, dass im Unterschied zu (1) die Beobachtungen von  $H''$  in der Sequenz vorkommen.

Neben Beobachtungen lassen sich in ähnlicher Weise auch Geheimnisse an bestimmten Ereignissen festmachen. So kann die Tatsache, dass  $A$  ein Angebot von  $H$  erhalten hat (also ein Geheimnis für alle Händler-Agenten außer  $H$  selbst), am Auftreten des Events  $receive_A(H, O)$  festgemacht werden.

Mit Hilfe von Sequenzen von Events lassen sich nicht nur Beobachtungen von Agenten sondern auch, genereller, das Verhalten von Systemen (wie z.B. von Multi-Agenten-Systemen) modellieren. Die möglichen Ausführungsfolgen eines Systems werden dazu als eine Menge  $Tr \subseteq E^*$  (wobei  $E$  die Menge der Events ist) von Ausführungsfolgen, sogenannten *Traces*, beschrieben. D. h. ein Trace  $\tau \in Tr$  ist eine Folge von Events, die ein mögliches Systemverhalten modelliert. Traces, in denen z.B.  $H'$  eine Anfrage von  $A$  bekommt, ohne dass vorher  $A$  eine Anfrage an  $H'$  geschickt hat, sind *nicht* in der Menge  $Tr$  der Virtual Mall enthalten, weil sie keinem möglichen Systemverhalten entsprechen.<sup>5</sup> Ebenso werden Traces ausgeschlossen, in denen etwa  $H'$  ein Angebot an  $A$  schickt, ohne vorher eine Anfrage von  $A$  bekommen zu haben. D. h. die Menge der möglichen Traces ist beschränkt auf die Traces, die aufgrund der Annahmen an die Umgebung der Agenten (z. B. die Kommunikation) und an das Verhalten der Agenten (ihr intern ablaufendes Programm) überhaupt möglich sind.

Die oben eingeführten Beobachtungen eines Agenten (oder von Gruppen von Agenten) während eines Systemablaufs ergeben sich als Projektionen des aktuellen Traces auf die Menge der Events  $V$ , die an der Schnittstelle des Agenten (oder an Schnittstellen von Agenten der Gruppe) geschehen. Zusammenfassend lassen sich die Beobachtungen eines Agenten  $H'$  durch eine Menge von sichtbaren Events  $V_{H'}$ , und die für ihn relevanten Geheimnisse durch eine Menge von Events  $C_{H'}$ , die vor  $H'$  geheim sind (d.h.  $H'$  darf sie nicht erfahren), beschreiben. In gleicher Weise lassen sich die Beobachtungen von Mengen von Agenten und die Geheimnisse durch zwei Mengen  $V$  und  $C$  von Events ausdrücken, die angeben, welche Events jeweils sichtbar bzw. geheim sind.

Offensichtlich ist es nicht gestattet, dass Events sowohl zu  $V$  als auch zu  $C$  gehören, d.h.  $C \cap V = \{\}$ . Alle Events, die weder sichtbar noch geheim sind, werden einer dritten Menge  $N$  zugeordnet. Somit ergibt sich eine disjunkte Partition  $(V, N, C)$ , die als *View* bezeichnet wird.<sup>6</sup>

### 4.3 Formalisierung von Sicherheitsanforderungen

Wir betrachten wieder die folgende Anforderung aus dem vorangegangenen Abschnitt:

Die Nachrichten, die ein Käufer-Agent an Händler (außer an  $H$ ) verschickt, dürfen nicht von Angeboten abhängen, die er von Händler  $H$  erhalten hat.

Dazu ist es ausreichend, die Beobachtung eines beliebigen Käufer-Agenten  $A$  zu betrachten. D. h. man betrachtet die Projektion der möglichen Traces aus  $Tr$  auf die Events, die an der Schnittstelle von  $A$  oder in  $A$  intern geschehen. Als Beispiel ist etwa folgende Sequenz möglich:

$$\langle receive_A(M, [H, H', H'']), send_A(H, R_1), receive_A(H, O), store\_offer_A(H, O), \quad (3) \\ send_A(H', R_1), send_A(H'', R_1), receive_A(H', O_2), store\_offer_A(H', O_2) \rangle.$$

In dieser Sequenz sind die für die Händler  $H'$  und  $H''$  beobachtbaren Events durch Unterstreichung und geheime Events durch eine gebrochene Unterstreichung hervorgehoben. Die beobachtbaren Events korrespondieren zu Geschehnissen an den Schnittstellen von  $H'$  und  $H''$ , wie z.B. der Event  $send_A(H', R_1)$  (ohne den der Event  $receive_H(A, R_1)$  nicht möglich ist). Wir werden der Einfachheit halber im Folgenden annehmen, der Event  $send_A(H', R_1)$  sei direkt für  $H'$  sichtbar.

In bezug auf die hier betrachtete Abhängigkeitseigenschaft, nehmen wir an, dass der (gebrochen unterstrichene) Event  $receive_A(H, O)$  für  $H'$  und  $H''$  ein geheimer Event sei. Damit  $H'$  und  $H''$  nicht auf ein

<sup>5</sup>Andererseits können Traces enthalten sein, in denen ein Senden einer Nachricht nicht von einem Empfangen gefolgt wird. Das ist z. B. sinnvoll, wenn ein System betrachtet wird, in dem Nachrichten verloren gehen können.

<sup>6</sup>Die Bezeichnung der Mengen in einer *view* sind aus dem Englischen motiviert.  $V$  steht für *visible* (sichtbar),  $C$  für *confidential* (vertraulich) und  $N$  für *non-visible/non-confidential* (nicht sichtbar/nicht vertraulich).

Geschehen dieses geheimen Events schließen können, dürfen die für  $H'$  und  $H''$  beobachtbaren Events nicht vom Geschehen des Events  $receive_A(H, O)$  abhängig sein. In anderen Worten, wäre der Event  $receive_A(H, O)$  nicht geschehen, so müssten die für  $H'$  und  $H''$  sichtbaren Events trotzdem möglich sein. Die Anforderung ist also, dass der Trace, der durch Streichen von  $receive_A(H, O)$  aus (3) entsteht,<sup>7</sup> d.h.

$$\langle receive_A(M, [H, H', H'']), send_A(H, R_1), \\ send_A(H', R_1), send_A(H'', R_1), receive_A(H', O_2), store\_offer_A(H', O_2)) \rangle \quad (4)$$

aus Sicht von  $H'$  und  $H''$  einem möglichen Systemverhalten entspricht. Ist dieser Trace tatsächlich möglich, so können  $H'$  und  $H''$  aus ihrer gemeinsamen Beobachtung

$$\langle send_A(H, R_1), send_A(H', R_1), receive_A(H', O_2) \rangle \quad (5)$$

nicht schließen, ob der Event  $receive_A(H, O)$  wirklich stattgefunden hat. Beide Ausführungsfolgen (d. h. mit und ohne den geheimen Event) sind ja möglich und  $H'$  und  $H''$  können nicht unterscheiden, welche der beiden stattgefunden hat, weil sich für beide Traces dieselbe Beobachtung für sie ergibt.

Betrachtet man diese Anforderung allgemeiner, so sagt sie: Wenn Traces mit geheimen Events möglich sind, dann muss nach Streichen der geheimen Events jeweils wieder ein möglicher Trace entstehen. Formal ist diese Anforderung nichts anderes als eine bestimmte Abschlusseigenschaft der Menge der möglichen Traces des Systems.

Diese Abschlusseigenschaft lässt sich wie folgt formalisieren:

$$\forall \alpha, \beta \in E^*, \forall c \in C_{H, H''}. ((\beta \cdot \langle c \rangle \cdot \alpha \in Tr_A \wedge \alpha|_{C_{H, H''}} = \langle \rangle) \\ \Rightarrow \exists \alpha' \in E^*. (\beta \cdot \alpha' \in Tr_A \wedge \alpha'|_{V_{H, H''}} = \alpha'|_{V_{H, H''}} \wedge \alpha'|_{C_{H, H''}} = \langle \rangle)) \quad (6)$$

Dabei ist  $V_{H, H''}$  die Menge der für  $H'$  oder  $H''$  sichtbaren Events und  $C_{H, H''}$  die Menge der für  $H'$  und  $H''$  geheimen Events. Weiterhin ist  $Tr_A$  die Menge aller möglichen Traces für Agent  $A$ ,  $\langle \rangle$  ist der leere Trace und  $\beta \cdot \alpha$  ist die Konkatenation der Traces  $\alpha$  und  $\beta$ . Schließlich ist  $\alpha|_X$  die Projektion des Traces  $\alpha$  auf die Menge von Events  $X \subseteq E$ , d. h. der Trace, der entsteht, wenn aus  $\alpha$  alle Events gestrichen werden, die nicht in der Menge  $X$  enthalten sind. Die Formel fordert, dass für jeden möglichen Trace  $\beta \cdot \langle c \rangle \cdot \alpha \in Tr_A$  ein möglicher Trace  $\beta \cdot \alpha' \in Tr_A$  existiert, der unter anderem die folgenden Eigenschaften hat:

- Der letzte, also am weitesten rechts stehende, geheime Event ist aus dem Trace gestrichen. In der Formel ist das  $c \in C_{H, H''}$  im Trace  $\beta \cdot \langle c \rangle \cdot \alpha$ .
- Die Beobachtungen von  $H'$  und  $H''$  sind unverändert, d. h.  $(\beta \cdot \langle c \rangle \cdot \alpha)|_{V_{H, H''}} = (\beta \cdot \alpha')|_{V_{H, H''}}$ .

Die Möglichkeit, dass  $\alpha'$  (auf kontrollierte Art und Weise) von  $\alpha$  abweichen darf, erlaubt es in dem oben angegebenen Beispiel (3), den internen (und für  $H'$  und  $H''$  nicht sichtbaren) Event  $store\_offer(H, m)$  zusammen mit dem geheimen Event  $receive_A(H, m)$  zu entfernen. Die durch (6) definierte Abschlusseigenschaft ist ein Beispiel für ein sogenanntes *Basic Security Predicate* (BSP).

#### 4.4 Verwendetes Framework: MAKS

Im letzten Abschnitt über die Formalisierung des Systems und dessen Sicherheitsanforderungen verzichteten wir weitgehend auf Formeln und vereinfachten einige Details, um ein besseres Verständnis für die grundlegende Methodik zu erreichen. Die bei der Formalisierung verwendeten Konzepte entstammen einem Framework für Informationsflusseigenschaften [5] namens *Modular Assembly Kit for Security Properties* (MAKS). Für den interessierten Leser geben wir kurz die wichtigsten Definitionen der in MAKS verwendeten Konzepte an und verweisen auf die sich durch die Verwendung des Frameworks eröffnenden Möglichkeiten sowie weitere Anwendungsgebiete. Weitere Details findet der interessierte Leser in der jeweils angebenen Literatur.

**Definition 1 (Event System).** Ein Event System ist ein Quadrupel  $(E, I, O, Tr)$ , wobei  $E$  die Menge der Events ist,  $I, O \subseteq E$  die Mengen der Eingabe- und Ausgabe-Events sind und  $Tr \subseteq E^*$  die Menge der möglichen Traces ist.

<sup>7</sup>Im Beispiel widerspricht der Event  $store\_offer_A(H, O)$  der Spezifikation des Käufer-Agenten, der ein Angebot nur speichert, wenn er es auch empfangen hat. Da dieser Event aber für  $H'$  und  $H''$  nicht sichtbar ist, darf er entfernt werden, vgl. die Formalisierung (6) der Eigenschaft für weitere Details.

**Definition 2 (View).** Eine View ist ein Tripel  $(V, N, C)$ , wobei  $V, N, C \subseteq E$  eine disjunkte Partition von  $E$  bilden.

**Definition 3 (Basic Security Predicate).** Ein Basic Security Predicate ist eine Abschlusseigenschaft auf Mengen von Traces, die parametrisch in einer View sind.

Dieses ist eine abstrakte Definition von BSPs. Ein Beispiel für ein konkretes BSP ist ((6)). Diese Formel entspricht der Definition von Backwards-Strict Deletion (BSD), einem speziellen BSP, das im Rahmen von *MAKS* verwendet wird. Es gibt verschiedene weitere BSPs, die in *MAKS* eingesetzt werden können um andere Formen von Nichtbeeinflussung auszudrücken.

**Definition 4 (Security Predicate).** Ein Security Predicate ist eine Menge  $\{BSP^1, \dots, BSP^n\}$  von Basic Security Predicates  $BSP^i$ . Ein Security Predicate gilt für eine View  $(V, N, C)$  und eine Menge von Traces  $Tr$  (geschrieben  $SP_{V,N,C}(Tr)$ ), wenn  $BSP^i_{V,N,C}(Tr)$  für alle  $i \in \{1, \dots, n\}$  gilt.

Die BSPs in einem Security Predicate sind also konjunktiv verknüpft, deshalb schreiben wir für ein Security Predicate  $SP$  auch  $BSP^1 \wedge \dots \wedge BSP^n$  (anstatt von  $\{BSP^1, \dots, BSP^n\}$ ).

Damit ergibt sich folgende Repräsentation von Informationsflusseigenschaften in *MAKS*.

**Definition 5 (Informationsflusseigenschaft).** Eine Informationsflusseigenschaft in *MAKS* ist ein Paar  $(VS, SP)$ , wobei  $VS$  eine Menge von Views und  $SP$  ein Security Predicate ist. Ein Event-System  $(E, I, O, Tr)$  erfüllt eine Informationsflusseigenschaft  $(VS, SP)$ , wenn  $SP_{V,N,C}(Tr)$  für jede View  $(V, N, C)$  in  $VS$  gilt.

An anderer Stelle wurde bereits gezeigt, dass sich die wichtigsten bekannten Informationsflusseigenschaften in *MAKS* repräsentieren lassen [5]. Beispiele für in *MAKS* repräsentierte Informationsflusseigenschaften sind Generalized Noninterference [10], Noninference [12] oder Separability [11]. Durch die uniforme Repräsentation erleichtert *MAKS* den Vergleich solcher Sicherheitseigenschaften aber auch andere Untersuchungen. Zum Beispiel wird die Herleitung von Verifikationstechniken (Stichwort Unwinding) durch *MAKS* extrem vereinfacht [6]. Ebenso lassen sich Kompositionalitätsresultate mit Hilfe von *MAKS* relativ leicht herleiten [7]. Diese Resultate können in der Praxis verwendet werden, um die Verifikation von Informationsflusseigenschaften zu vereinfachen. Mit Hilfe von Kompositionalitätsresultaten kann die Verifikationsaufgabe für ein komplexes Gesamtsystem auf Verifikationsaufgaben für Systemkomponenten heruntergebrochen werden. Lässt sich die Verifikationsaufgabe nicht weiter auf Komponenten herunterbrechen, dann können Unwinding-Resultate eingesetzt werden, um die Verifikation der Sicherheitsanforderungen des gesamten Systems zu erleichtern. Beispiele für Anwendungen von Informationsflusseigenschaften, bei denen diese Vorteile von *MAKS* genutzt wurden, finden sich unter [8,9]. Weitere Anwendungen von Informationsflusseigenschaften finden sich in [3,14].

## 5. Umsetzung von Vertraulichkeitseigenschaften

Im Folgenden betrachten wir einen einfachen Systemablauf, in dem der Käufer-Agent nachdem er alle Angebote von den Händler-Agenten eingeholt hat, das beste Angebot der Tabelle auswählt und dem entsprechenden Händler-Agenten  $H$  einen Kaufauftrag  $send_A(H, buy(A, O))$  schickt. Nach Verstreichen einer gewissen Frist bemerkt ein Händler  $H'$ , der ebenfalls ein geeignetes Angebot abgegeben hat, daß ein anderer Händler den Zuschlag bekommen haben muss, da er selbst keinen Auftrag erhalten hat.<sup>8</sup> Betrachtet man diesen Verhandlungsverlauf aus Sicht des Händlers  $H'$ , so müsste gemäß unserer Abschlusseigenschaft ((6) neben dem möglichen Trace

$$\langle \dots, receive_A(H, O), store\_offer_A(H, O), receive_A(H', O_1), store\_offer_A(H', O_1) \quad (7) \\ send_A(H, buy(A, O)), timeout_{H'}(O_1) \rangle$$

<sup>8</sup>Wir modellieren das Ereignis, dass der Händler  $H'$  das Verstreichen der Frist seines Angebots  $O_1$  an  $A$  bemerkt durch einen Event  $timeout_{H'}(O_1)$ . Verschiedene andere Modellierungen sind möglich. Z.B. könnte das Verstreichen von Zeiteinheiten von durch *tick*-events modelliert werden [4], wobei das Verstreichen der Frist einer bestimmten Zahl solcher *tick*-events entspräche. Für unsere Zwecke ist die gewählte Variante ausreichend.

in  $Tr_A$  auch ein weiterer Trace möglich sein, der aus (7) entsteht, indem die geheimen Events gestrichen werden, und der für  $H'$  zu denselben Beobachtungen führt. Durch einfaches Streichen des geheimen Events  $receive_A(H, O)$  entstünde

$$\langle \dots, \text{store\_offer}_A(H, O), \text{receive}_A(H', O_1), \text{store\_offer}_A(H', O_1), \text{send}_A(H, \text{buy}(A, O)), \text{timeout}_H(O_1) \rangle \quad (8)$$

Dies ist allerdings kein möglicher Trace in  $Tr_A$ , weil der Käufer-Agent darin ein Angebot  $O$  von  $H$  speichert und später eine Ware bei  $H$  kauft, ohne dafür jemals ein Angebot von  $H$  erhalten zu haben. Die Frage ist nun, ob es für die Beobachtungen von  $H'$  eine Erklärung ohne den geheimen Event  $receive_A(H, O)$  gibt. Eigenschaft (6) lässt zu, dass dazu der Trace nach dem entfernten Event kontrolliert korrigiert werden darf: Die sichtbaren Events müssen aber in der richtigen Reihenfolge erhalten bleiben, und es dürfen weder sichtbare noch geheime Events eingefügt werden. Da  $\text{store\_offer}_A(H, O)$  weder geheim noch sichtbar ist, dürfen wir diesen Event entfernen (vgl. Fußnote 7). Auch der Event  $\text{send}_A(H, \text{buy}(A, O))$  darf entfernt werden. Somit ergäbe sich folgender Trace

$$\langle \dots, \text{receive}_A(H', O_1), \text{store\_offer}_A(H', O_1), \text{timeout}_H(O_1) \rangle \quad (9)$$

Allerdings ist dieses wiederum kein möglicher Trace für unsere Virtual Mall. Da der Käufer-Agent mindestens ein passendes Angebot erhalten hat (das von  $H'$ ),  $H'$  aber keinen Zuschlag bekommen hat, muß irgendein anderer Händler-Agent ein besseres Angebot abgegeben haben. Das Abgeben eines solchen Angebots entspricht aber einem geheimen Event und geheime Events dürfen nicht zur Korrektur in ein Trace eingefügt werden (vgl. ((6)). Unsere (formalisierte) Sicherheitseigenschaft gilt also für dieses System nicht (zu Recht).

Das obige Beispiel zeigt, dass in dem zugrunde gelegten einfachen Szenario die von uns formulierten Sicherheitsinteressen des Händlers  $H$  verletzt werden. Betrachten wir den kritischen Fall etwas detaillierter. Die Anforderung besagt, dass die Beobachtungen des Händlers  $H'$  über den Käufer (also die Anfrage und der nicht erfolgte Kauf bei  $H'$ ) auch mögliche Beobachtungen wären, falls ein tatsächlich erfolgtes Angebot eines anderen Händlers  $H$  nicht gemacht worden wäre. Für alle Angebote außer dem besten Angebot ist diese Anforderung erfüllt: sie beeinflussen die Beobachtungen von  $H'$  nicht. Das beste Angebot beeinflusst jedoch die Beobachtungen von  $H'$ , insbesondere wenn das Angebot von  $H'$  das zweitbeste war: Wäre das bessere Angebot nicht tatsächlich abgegeben worden, so hätte  $H'$  den Zuschlag bekommen. Er kann also aus seinen Beobachtungen (speziell dem nicht erfolgten Kauf) zweifelsfrei schließen, dass ein Angebot eines anderen Händlers abgegeben wurde. Dieser Schluss beruht auf mehreren Annahmen.

1. Der Händler weiß, dass sein Angebot den Anforderungen des Käufers entsprochen hat.
2. Der Händler weiß, dass sich der Käufer für eines der Angebote entscheidet und kauft, wenn mindestens eines der Angebote seinen Anforderungen entspricht.
3. Der Händler weiß nach Verstreichen der Angebotsfrist, dass der Käufer-Agent eine anderweitige Kaufentscheidung bereits getroffen hat.

Es bieten sich nun mehrere Möglichkeiten an, die Sicherheit des Systems dadurch zu gewährleisten, dass man die Randbedingungen des Systems so ändert, dass mindestens eine dieser Annahmen nicht mehr gilt. Nach einer solchen Änderung kann der Händler  $H'$  die oben skizzierten Schlüsse nicht mehr ziehen.

1. Käufer könnten die Kaufentscheidung unendlich lange verzögern, so dass ein Händler nie weiß, ob eine Kaufentscheidung bereits gefallen ist oder nicht. Damit wäre die dritte Annahme nicht mehr gültig. Diese Möglichkeit ist allerdings funktional nicht sinnvoll, da der Sinn und Zweck des gesamten Szenarios aus Sicht des Käufers darin besteht, in endlicher Zeit eine Kaufentscheidung zu treffen. Er wäre sicherlich nicht bereit, zur Wahrung der Sicherheitsinteressen der Händler auf diese Funktionalität zu verzichten. Diese Möglichkeit scheidet also aus.
2. Ähnlich ist die Möglichkeit zu bewerten, dass Käufer nach der positiven Entscheidung für ein bestimmtes Angebot zufällig entscheiden, ob sie tatsächlich kaufen oder nicht. D.h. die zweite Annahme wird ungültig. Aber auch hier wäre der Käufer gezwungen, für ihn gute Angebote auszu-

schlagen, um die Sicherheitsinteressen der Händler zu gewährleisten – eine unrealistische Anforderung.

3. Es könnte Händler geben, die einerseits kein Interesse daran haben, dass ihre Angebote vertraulich behandelt werden, und andererseits aber selbst ihre Angebote nicht allgemein bekannt machen. Die Beobachtungen von  $H'$  wären dann immer durch die Abgabe eines Angebotes eines solchen Händlers erklärbar. Diese Abgabe ist in diesem Fall nicht vertraulich und deswegen unkritisch.<sup>9</sup> Auch diese Möglichkeit scheidet aus praktischen Erwägungen aus, weil es für einen Händler keine Motivation gibt, auf seine Interessen freiwillig zu verzichten.
4. Wenn ein Händler nach der Abgabe eines zur Anfrage passenden Angebotes nicht weiß, ob sein Angebot tatsächlich allen relevanten Anforderungen genügt, dann wäre die erste Annahme nicht mehr gültig. Die Beobachtung von  $H'$  wäre in diesem Fall auch ohne Abgabe eines anderen, besseren Angebotes dadurch erklärbar, dass sein Angebot für den Käufer nicht akzeptabel war und es deswegen nicht zu einem Kauf gekommen ist. Er kann deswegen aus seinen Beobachtungen nicht zweifelsfrei auf die Abgabe eines anderen (besseren) Angebotes schließen. Eine mögliche Realisierung dieser Lösung wäre, dass Käufer-Agenten nur unvollständige Anfragen stellen dürfen, die einige, aber nicht alle Anforderungen an die Ware umfassen. Durch die erzwungene Unvollständigkeit der Anfrage ist es für den Händler nicht mehr nachvollziehbar, aber er keinen Zuschlag bekommen hat, weil sein Angebot nicht adäquat war oder weil ein Konkurrent ein besseres Angebot gemacht hat. D.h. er kann aus seinen Beobachtungen nicht schließen, dass andere Händler Angebote abgegeben haben müssen. Eine weitere mögliche Realisierung dieser Lösung wäre es, wenn die Anfrage des Käufer-Agenten zwischen dem Zeitpunkt, zu dem das Angebot eingeholt wurde, und dem Zeitpunkt der Kaufentscheidung nachträglich verändert werden kann. Dadurch könnte z.B. ein weiteres, dem Händler unbekanntes Ausschlusskriterium vom Käufer hinzugefügt werden, durch das die angebotene Ware zum Zeitpunkt der Kaufentscheidung nicht mehr adäquat ist. Schließlich ist auch noch denkbar, dass vor dem Abschluss eines Kaufes der Benutzer des Käufer-Agenten noch mal explizit gefragt wird, ob er das beste Angebot (oder ein anderes) wirklich annehmen will, oder ob er auf einen Kauf trotz passender Angebote verzichtet (z.B. weil er die Ware doch nicht haben will).

Es gibt also mehrere Möglichkeiten, die Sicherheitslücke in der Virtual Mall zu verhindern. Allerdings scheint aus den angegebenen Gründen nur die vierte Möglichkeit wirklich praktikabel. Für diese Lösung gibt es mehrere mögliche Realisierungen, wobei die drei von uns angedeuteten Realisierungen, alle praktikabel erscheinen.

## 6. Zusammenfassung

In diesem Beitrag haben wir aufgezeigt, wie Sicherheitsanforderungen an ein Multi-Agenten-System formal mit Hilfe von Informationsflusseigenschaften überprüft werden können. Hierzu haben wir zunächst aus den verschiedenen Sicherheitsinteressen der Systembenutzer die globalen Sicherheitsanforderungen an das Gesamtsystem (der Virtual Mall) abgeleitet und haben diese danach auf lokale Sicherheitsanforderungen an die einzelnen Agenten reduziert. Unser Fokus war dabei auf Anforderungen an die Vertraulichkeit, die wir als Abhängigkeitseigenschaften informell aber präzise formuliert haben. Wie solche informellen Beschreibungen mit Informationsflusseigenschaften formalisiert werden können, wurde allgemein angegeben. Dabei haben wir Konzepte aus einem allgemeinen Framework namens *MAKS* [5] zur Behandlung von Informationsflusseigenschaften eingesetzt. Durch unser schrittweises Vorgehen wurde die Formalisierung der Sicherheitseigenschaften deutlich erleichtert. Weiterhin haben wir aufgezeigt, wie Sicherheitslücken mit Hilfe der formalisierten Sicherheitseigenschaften entdeckt werden können und haben verschiedene Möglichkeiten angegeben, um unser Beispielsystem sicher zu machen.

Obwohl unsere Betrachtungen in diesem Artikel auf ein konkretes Szenario aus dem Comparison Shopping beschränkt waren, sind wir überzeugt, dass sich das vorgeschlagene schrittweise Vorgehen bei der Formalisierung von Sicherheitseigenschaften auch für andere Systeme gewinnbringend anwenden lässt.

Durch die Verwendung eines bekannten, generischen Frameworks ergeben sich verschiedene Möglichkeiten, die in diesem Artikel skizzierte formale Betrachtung zu erweitern. So wäre es interessant, die

---

<sup>9</sup>Technisch bedeutet dies, dass diese Angebotsabgaben Events aus  $N$  (und nicht aus  $C$ ) sind.

Rolle der Plattform und die Sicherheitsanforderungen an die Plattform – etwa bezüglich der Kommunikation von Agenten – genauer zu untersuchen. Ein anderer interessanter Ansatz wäre es, die durch MAKS zur Verfügung gestellten Kompositionalitätsresultate [7] einzusetzen, um auf modulare Art und Weise Sicherheitseigenschaften des gesamten Multi-Agenten-Systems aus den Eigenschaften der einzelnen Agenten formal abzuleiten.

## Literatur

- [1] Bowen Alpern and Fred B. Schneider. Defining Liveness. *Information Processing Letters*, 21: 181–185, 1985. North-Holland.
- [2] A. Chavez and P. Maes. An Agent Marketplace for Buying and Selling Goods. In *Proceedings of the 1st International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*, London, UK, 1996.
- [3] Riccardo Focardi, Anna Ghelli, and Roberto Gorrieri. Using Non Interference for the Analysis of Security Protocols. In *Proceedings of DIMACS Workshop on Design and Formal Verification of Security Protocols*. DIMACS Center, Rutgers University, 1997.
- [4] Riccardo Focardi, Roberto Gorrieri, and Fabio Martinelli. Information Flow Analysis in a Discrete-Time Process Algebra. In *Proceedings of the 13th IEEE Computer Security Foundations Workshop*, pages 170–184, Cambridge, UK, 2000.
- [5] Heiko Mantel. Possibilistic Definitions of Security – An Assembly Kit. In *Proceedings of the 13th IEEE Computer Security Foundations Workshop*, pages 185–199, Cambridge, UK, 2000.
- [6] Heiko Mantel. Unwinding Possibilistic Security Properties. In *European Symposium on Research in Computer Security (ESORICS)*, LNCS 1895, pages 238–254, Toulouse, France, 2000.
- [7] Heiko Mantel. On the Composition of Secure Systems. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 88–101, Oakland, CA, USA, 2002.
- [8] Heiko Mantel and Andrei Sabelfeld. A Unifying Approach to the Security of Distributed and Multi-Threaded Programs. *Journal of Computer Security (JCS)*, 2002. to appear.
- [9] Heiko Mantel, Axel Schairer, Matthias Kabatnik, Michael Kreutzer, and Alf Zugenmaier. Using Information Flow Control to Evaluate Access Protection of Location Information in Mobile Communication Networks. Technical Report 159, Institut für Informatik, Universität Freiburg, August 2001.
- [10] Daryl McCullough. Specifications for Multi-Level Security and a Hook-Up Property. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 161–166, Oakland, CA, 1987.
- [11] John D. McLean. A General Theory of Composition for Trace Sets Closed under Selective Interleaving Functions. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 79–93, Oakland, CA, 1994.
- [12] Colin O’Halloran. A Calculus of Information Flow. In *Proceedings of the European Symposium on Research in Computer Security (ESORICS)*, pages 147–159, Toulouse, France, 1990.
- [13] Ina Schaefer. Secure Mobile Multiagent Systems in Virtual Marketplaces. A Case Study on Comparison Shopping. Research Report RR-02-02, Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI GmbH, 2002.
- [14] Gerhard Schellhorn, Wolfgang Reif, Axel Schairer, Paul Karger, Vernon Austel, and David Toll. Verification of a Formal Security Model for Multiapplicative Smart Cards. In *European Symposium on Research in Computer Security (ESORICS)*, LNCS 1895, pages 17–36, Toulouse, France, 2000.
- [15] M. Tsvetovaty, B. Mobasher, M. Gini, and Z. Wieckowski. MAGMA: An Agent Based Virtual Market Place for Electronic Commerce. *Applied Artificial Intelligence*, 1997.