

Seminararbeit

Artefakt-basierte Modellierung von Geschäftsprozessen

Katrin Holterbork
25. Juni 2012

Katrin Holterbork
Katrin.Holterbork@udo.edu.de
Matrikelnummer: 146041
Studiengang: Master Informatik

Ausgewählte Themen des Modellbasierten Sicherheits-Engineerings
Thema: Artefakt-basierte Modellierung von Geschäftsprozessen

Eingereicht: 25. Juni 2012

Betreuer: Andreas Schmitz

Prof. Dr. Jan Jürjens Lehrstuhl 14 Software Engineering
Fakultät Informatik
Technische Universität Dortmund
Otto-Hahn-Straße 14
44227 Dortmund

Ehrenwörtliche Erklärung

Ich erkläre hiermit ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Dortmund, den 25. Juni 2012

Katrin Holterbork

Inhaltsverzeichnis

Abbildungsverzeichnis	vii
1 Einleitung	1
2 Modellierung von Geschäftsprozessen	3
2.1 Artefakt-basierte Modellierung	3
2.1.1 Business Artefakte	3
2.1.2 Modellierungs-Beispiele	6
2.2 Ereignis-basierte Modellierung	8
2.2.1 Einführung	8
2.2.2 Beispiel in BPMN	9
2.3 Vergleich der Modellierungsarten	10
3 Analyse der Artefakt-basierten Modellierung	13
3.1 Formales Modell	13
3.2 Analyse und Ergebnisse	17
4 Fazit	19

Abbildungsverzeichnis

2.1	Inhalt eines Artefakts [NC03]	4
2.2	Lebenszyklus einer Gast Abfertigung (nach [NC03])	6
2.3	IT-Service Provider [BGH ⁺ 07]	8
2.4	Beispiel Guest Check in der BPMN	10
3.1	Beispiel Artefaktklasse und Artefakt [BGH ⁺ 07]	14
3.2	Beispiel Service [BGH ⁺ 07]	15
3.3	Beispiel Business-Regeln [BGH ⁺ 07]	16

1 Einleitung

Die Modellierung von Geschäftsprozessen ist eine wichtige Herausforderung für Unternehmen. Geschäftsprozesse sorgen dafür, dass Unternehmen dem wachsenden Druck an den Märkten standhalten und flexibel auf Änderungen reagieren können. Dafür sind gut modellierte Geschäftsprozesse notwendig.

Am weitesten verbreitet ist dabei die Ereignis-basierte Modellierung. Im Gegensatz zur Ereignis-basierten Modellierung von Geschäftsprozessen, liegt der Fokus der Artefakt-basierten Modellierung auf den bewegbaren Daten, die während des Prozesses manipuliert werden und nicht bei den Ereignissen und Aktivitäten, die während des Prozesses ablaufen.

Motivation für den Artefakt-basierten Ansatz ist, dass jeder Prozess Informationen aufzeichnen muss und dass diese Aufzeichnungen alle für den Prozess relevanten Daten enthalten sollten. Durch die Fokussierung auf diese Daten wird die Flexibilität für die Business Prozesse vergrößert, da nicht die Ereignisse und Aktivitäten, also der Ablauf modelliert wird, sondern das Ergebnis in Form von Business Artefakten.

Der Aufbau dieser Seminararbeit sieht wie folgt aus. Zunächst werden in Kapitel 2 die Artefakt-basierte Modellierung von Geschäftsprozessen, sowie die Ereignis-basierte Modellierung von Geschäftsprozessen vorgestellt. Dabei liegt der Schwerpunkt auf der Artefakt-basierten Modellierung. Beide Modellierungsarten werden in dem Kapitel anschließend verglichen und ihre Vor- und Nachteile herausgestellt. In Kapitel 3 wird eine Analyse der Artefakt-basierten Modellierung vorgestellt. Dabei liegt der Schwerpunkt auf der formalen Darstellbarkeit der Artefakt-basierten Modellierung. Zum Schluss werden in Kapitel 4 ein Fazit verfasst und weitere Forschungsbereiche in Hinsicht auf die Artefakt-basierte Modellierung diskutiert.

2 Modellierung von Geschäftsprozessen

In diesem Kapitel wird die Modellierung von Geschäftsprozessen erläutert. Es wird zunächst in Unterkapitel 2.1 der Artefakt-basierte Ansatz zur Modellierung von Geschäftsprozessen vorgestellt. Danach wird in Unterkapitel 2.2 kurz der Ereignis-basierte Ansatz zur Modellierung von Geschäftsprozessen erläutert. Beide Modellierungsarten werden anschließend in Unterkapitel 2.3 verglichen und ihre Vor- und Nachteile werden erörtert.

2.1 Artefakt-basierte Modellierung

Zunächst wird die Artefakt-basierte Modellierung von Geschäftsprozessen beschrieben. In Unterkapitel 2.1.1 wird erläutert, was unter einem Business Artefakt zu verstehen ist. Anschließend werden in Unterkapitel 2.1.2 zwei Beispiele zur Artefakt-basierten Modellierung gezeigt.

2.1.1 Business Artefakte

Die Hauptelemente der Artefakt-basierten Modellierung sind die Business Artefakte oder kurz Artefakte. Nigam und Caswell beschreiben diese Artefakte als konkrete, identifizierbare und sich selbst beschreibende Stücke an Informationen, die von Geschäftsleuten dafür genutzt werden können ein Business am laufen zu halten [NC03]. Ein Artefakt kapselt alle Informationen, die für den Prozess benötigt werden, an einem Ort. Konkret bedeutet, dass das Artefakt genau beschrieben ist. Identifizierbar ist jedes Artefakt durch eine eindeutige Identität. Der Inhalt eines Artefakts sorgt dafür, dass es selbst beschreibend ist.

Es darf keine zwei Artefakte mit der selben Identität geben, allerdings dürfen Artefakte identischen Inhalt besitzen und sich ausschließlich in ihrer Identität unterscheiden [NC03]. Zu der Bedingung der eindeutigen Identität kommt noch die Bedingung, dass die Identität eines Artefaktes nicht änderbar ist. Sie muss gleich bleiben und zu jeder Zeit eindeutig feststehen. Bei einer Teilung entstehen zwei Artefakte mit der selben Identität. Da das nicht erlaubt ist und das Abändern einer der Identitäten ebenfalls nicht erlaubt ist, folgt daraus, dass ein Artefakt nicht in zwei oder mehrere Artefakte geteilt werden darf.

Der Inhalt eines Artefakts kann und soll hingegen veränderbar sein. Dargestellt werden kann er z. B. durch Namen-Werte-Paare, wie in Abbildung 2.1 beispielhaft dargestellt ist [NC03]. Diese Werte werden im weiteren als Attribute bezeichnet. Verändert werden können zum einen die Attribute des Artefakts selbst, oder das Artefakt an sich, indem neue Attribute, also Namen-Werte-Paare, hinzugefügt wer-

```
guest-check {  
  ID 123  
  context ()  
  customer (number 3)  
  store (ID(55) server (2))  
  item (desc HamB price 2.57  
    cooked "13:23 04/17/1998")  
  delivered "13:26 04/17/1998"  
  tax 0.33  
  tender (total 2.90  
    cash 20.00 coupon 1.00  
    change 18.10)  
}
```

Abbildung 2.1: Inhalt eines Artefakts [NC03]

den. Der Inhalt eines Artefakts kann auch kopiert werden und z. B. einem anderen Artefakt hinzugefügt werden.

Um Geschäftsprozesse mit Hilfe von Artefakten modellieren zu können, werden noch weitere Elemente neben den Artefakten benötigt. Es gibt unterschiedliche Möglichkeiten für die Modellierung der Geschäftsprozesse. Zwei davon werden hier betrachtet. Dabei handelt es sich um eine Technik von Nigam und Caswell aus dem Paper [NC03] und eine Technik aus dem Paper [BGH⁺07], auf die sich die spätere Analyse in Kapitel 3 bezieht.

Nigam und Caswell stellen in ihrem Paper die von ihnen entwickelte Technik OpS (Operational Specification) vor. Die Technik soll einen Prozess sowohl formal beschreiben können als auch intuitiv zu verstehen sein [NC03]. Die zweite Modellierungstechnik aus [BGH⁺07] baut auf OpS auf, daher werden hier nur die Unterschiede zu OpS erläutert.

Modelliert wird der Lebenszyklus eines Artefakts. Nigam und Caswell nutzen dazu in der OpS neben den Artefakten noch Services und Repositories. Unter Services sind Funktionen zu verstehen, die an einem Ort gekapselt sind und Artefakte verändern können. Repositories sind Orte an denen Artefakte aufgehoben werden und auf weitere Verarbeitung warten können.

Zunächst werden hier grundsätzliche Eigenschaften von Services und Repositories zusammengestellt. Nigam und Caswell stellen diese Eigenschaften in sieben Axiomen zusammen [NC03]. Diese lauten wie folgt:

1. Das Ziel eines Service ist es eine Aktion durchzuführen und die Ergebnisse an ein, oder mehrere Artefakte zu berichten.
2. Ein Service wird entweder durch das Empfangen eines Artefakts ausgelöst oder durch einen expliziten Trigger-Mechanismus.
3. Ein Service transformiert ein oder mehrere Artefakte in seinem Zuständigkeitsbereich. Das kann das Hinzufügen von Inhalten oder das Verändern dieser sein.

Der Service kann Inhalte verschiedener Artefakte, die sich in seiner Zuständigkeit befinden, unter den Artefakten austauschen.

4. Artefakte werden von Services entweder spontan empfangen oder explizit angefordert.
5. Nach dem ein Service beendet ist, werden alle Artefakte wieder ausgesandt oder abgelegt.
6. Artefakte können in ein Repository eingefügt werden.
7. Ein Repository kann auf Anfragen nach Artefakten oder deren Inhalt antworten. Es kann Artefakte aber nicht spontan aussenden.

Des Weiteren werden noch drei Ablauf-Axiome vorgestellt, die sich mit dem Übergang der Artefakte von einem Ort zum anderen befassen [NC03]. Diese Axiome lauten wie folgt:

1. Eine Ablauf-Verbindung ist eine direkte Verbindung zwischen einem Start- und einem Zielort.
2. Eine Ablauf-Verbindung stellt eine sichere Verbindung da. Das Artefakt wird während der Übertragung nicht verändert.
3. Bei einer Verbindung zwischen einem Service und einem Repository hat die Verbindung eine Anfrage-Antwort Form. Ein Service der ein Artefakt anfordert bekommt also ein oder mehrere Artefakte oder eine Fehlermeldung zurück.

Zur Modellierung wird zunächst ein Schlüsselartefakt bestimmt, dessen Lebenszyklus vom Erzeugen bis zum Ende modelliert wird. In der OpS-Darstellung befinden sich Artefakte immer an einem bestimmten Ort. Das ist entweder ein Repository, wo sie auf weitere Verarbeitung warten, oder ein Service, wo sie verändert werden. Auf welche Weise ein Service ein Artefakt verändert hängt von den Attributen und deren Werten ab.

Die zweite Modellierungsvariante verzichtet auf Repositories und modelliert die Artefakte stattdessen mit einem Zustandsautomaten. Jedes Artefakt befindet sich zu jeder Zeit in einem bestimmten Zustand und nicht an einem Ort. Eine Zustandsänderung kann durch einen Service ausgelöst werden. Ansonsten agieren die Services wie bei der OpS. Sie verändern die Attribute eines Artefakts oder fügen neue Attribute hinzu. Allerdings werden konkret *Business-Regel* definiert, welche besagen, unter welchen Bedingungen ein Service auf welche Weise agiert. Diese Business-Regeln entscheiden, anhand des Zustandes und der Attribut-Werte von Artefakten, welche Attribute der Service wie verändert und ob er eine Zustandsänderung des Artefakts auslöst. Eine formale Beschreibung solcher Business-Regeln ist in Kapitel 3 zu finden.

Als nächstes werden zwei Beispiele zu den Modellierungsvarianten vorgestellt.

2.1.2 Modellierungs-Beispiele

Hier wird zunächst ein Beispiel zur Modellierung mit OpS und anschließend ein weiteres Beispiel zur Modellierung mit der Variante aus [BGH⁺07] beschrieben.

Gast Abfertigung (OpS) Das erste Beispiel zeigt die Modellierung eines Geschäftsprozesses, der die Abfertigung eines Gastes in einem Restaurant darstellt. In der OpS werden Services als Rechtecke mit abgerundeten Kanten dargestellt. Repositories werden durch Kreise veranschaulicht. In dem Beispiel in Abbildung 2.2 gibt es vier Services, sechs Repositories und fünf unterschiedliche Typen von Artefakten, wobei das Schlüsselartefakt das *Guest Check* Artefakt ist.

In dem Beispiel kommt ein Kunde in das Restaurant und wählt Speisen anhand einer Karte und einer Tageskarte aus. Die Wahl wird an die Küche weitergeleitet und das Essen zubereitet. Fertig bediente Gäste zahlen ihre Rechnung und die Bilanz des Geschäftes wird aktualisiert.

Wenn ein Kunde den Laden betritt wird der Service *Create Guest Check* ausgelöst. Ein weißer Kreis zeigt eine Stelle an, an der ein Service angestoßen wird. Dies geschieht entweder durch das Empfangen eines Artefakts oder durch das Empfangen von Inhalten, was an dieser Stelle der Fall ist. Der Service erstellt ein neues *Guest Check* Artefakt und schickt es ins *Active Guest Checks* Repository. Das Aussenden des Artefakts ist durch einen weißen Kreis mit einem X dargestellt. Dies geschieht immer dann, wenn ein Service abgeschlossen wird.

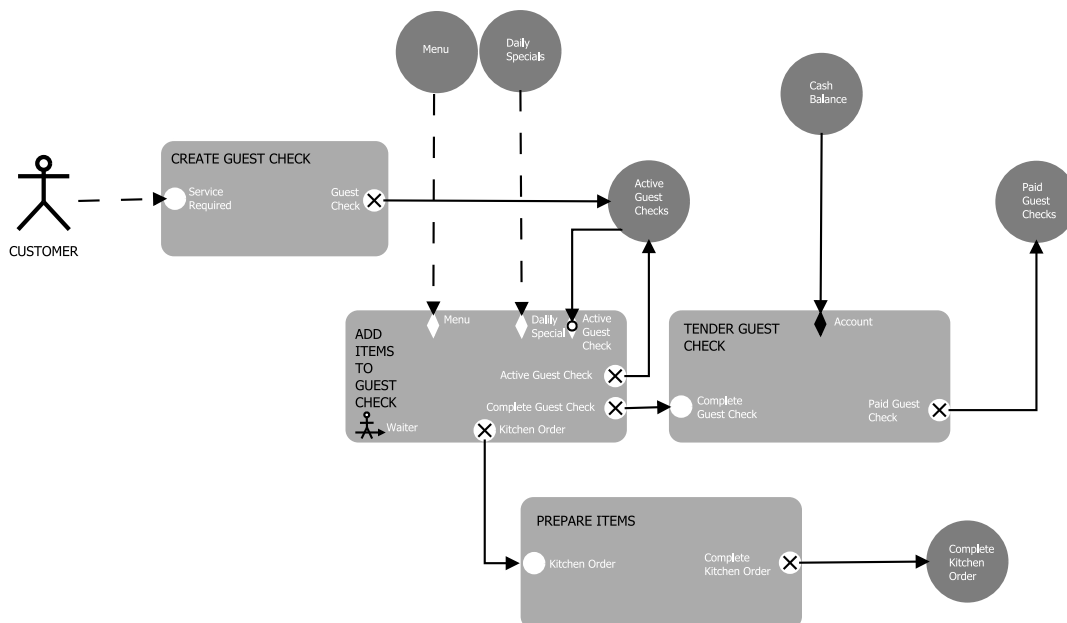


Abbildung 2.2: Lebenszyklus einer Gast Abfertigung (nach [NC03])

Der Service *Add Items to Guest Check* wird durch einen Gast ausgelöst, was durch das Symbol in der unteren linken Ecke gekennzeichnet ist. Der Service fordert das passende *Guest Check* Artefakt an und empfängt es. Dies ist durch eine weiße Raute mit Kreis modelliert. Zusätzlich fordert der Service noch den Inhalt der Artefakte *Menu* und *Daily Specials* an, was durch eine weiße Raute ohne Kreis modelliert ist. Der Inhalt der Artefakte reicht in diesem Fall, da sie vom Service nicht verändert werden sollen. Der Kunde wählt sein Essen aus und die neuen Informationen werden dem *Guest Check* Artefakt hinzugefügt. Es wird zurück in das *Active Guest Checks* Repository gesendet. Außerdem wird ein Artefakt *Kitchen Order* erstellt und an den Service *Prepare Items* gesendet. Damit ist der Service *Add Items to Guest Check* beendet.

Der Service *Prepare Items* wird durch das Eintreffen des Artefakts *Kitchen Order* ausgelöst. Er bearbeitet die Bestellung und sendet das Artefakt in das *Complete Kitchen Order* Repository, sobald die Bestellung vollständig erledigt ist.

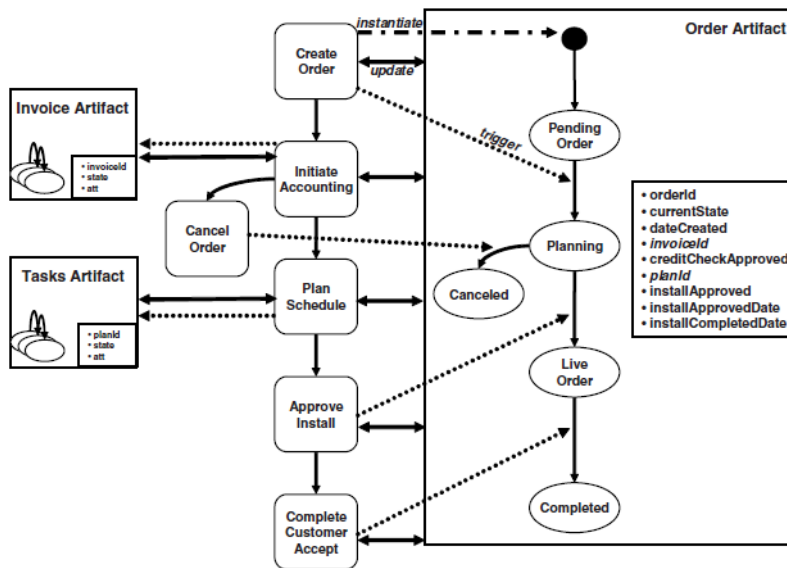
Der Service *Add Items to Guest Check* kann solange vom Wartenden ausgelöst werden, bis dieser fertig mit seinem Essen ist und das *Guest Check* Artefakt somit als abgeschlossen markiert wird. Das Artefakt wird dann nicht zurück ins Repository gesendet, sondern an den Service *Tender Guest Check* weitergeleitet.

Durch das Eintreffen des Artefakts wird dieser Service ausgelöst. Die Bezahlung wird abgehandelt und das *Guest Check* Artefakt als bezahlt markiert und nach Abschluss des Service in das *Paid Guest Checks* Repository gesendet. Zuvor fordert der Service noch das *Chash Balance* Artefakt an und aktualisiert die Bilanz. Dieses Artefakt wird nach der Aktualisierung direkt zurück in sein Repository gesendet. Dargestellt ist das Anfordern, Aktualisieren und Zurücksenden durch eine schwarze Raute. Damit ist der Service abgeschlossen und der Lebenszyklus des Artefakts *Guest Check* beendet.

IT-Service Provider In dem zweiten Beispiel geht es um einen IT Service Provider, der Service-Anfragen von einem Hotel bearbeitet. In Abbildung 2.3 ist der Geschäftsprozess modelliert.

Es gibt die drei Artefakte *Order*, *Invoice* und *Tasks*. Dabei ist das *Order* Artefakt das Schlüsselartefakt des Beispiels. Die Artefakte werden mit ihren Attributen und einem endlichen Zustandsautomaten dargestellt. Sie interagieren mit den Services, welche außerhalb der Artefakte als Business-Prozess dargestellt werden.

Der Service *Create Order* instantiiert das Artefakt *Order*. Zunächst befindet sich das Artefakt in dem Zustand *Pending Order*. Nachdem der Service *Create Order* das Artefakt erzeugt und einige Informationen hinzugefügt hat, löst er eine Zustandsänderung in den Zustand *Planning* aus. Durch definierte Business-Regeln wird der Service *Initiate Accounting* angestoßen. Der Service agiert mit dem Artefakt *Invoice*

Abbildung 2.3: IT-Service Provider [BGH⁺07]

und dem Artefakt *Order*. Durch die Ergebnisse und definierte Business-Regeln wird entschieden, ob weiter geplant oder die Anfrage abgebrochen wird. Soll sie abgebrochen werden, wird der Service *Cancel Order* ausgelöst und sorgt für eine Zustandsänderung des *Order* Artefakts zu *Canceled*. Der Lebenszyklus des Artefakts ist damit beendet, da ein Endzustand erreicht wird. Wird die Planung fortgeführt, so wird das *Order* Artefakt aktualisiert und als nächstes der Service *Plan Schedule* ausgelöst, welcher mit den *Tasks* und *Order* Artefakten interagiert. Das *Order* Artefakt wird um einen Business-Plan ergänzt und der Service *Approve Install* wird ausgeführt. Der Service löst eine Zustandsänderung des *Order* Artefakts zu *Live Order* aus und aktualisiert das Artefakt. Wenn der Service *Approve Install* abgearbeitet ist, löst der Service *Complete Customer Accept* wieder eine Zustandsänderung des *Order* Artefakts zu *Completed* aus und aktualisiert die Daten des Artefakts. Damit ist ein Endzustand des Artefakts eingetreten und der Lebenszyklus des *Order* Artefakts zu ende durchlaufen.

2.2 Ereignis-basierte Modellierung

In diesem Unterkapitel wird die Ereignis-basierte Modellierung von Geschäftsprozessen beschrieben. In Unterkapitel 2.2.1 wird eine kurze Einführung in die Ereignis-basierte Modellierung gegeben. In Unterkapitel 2.2.2 wird eins der Beispiele aus Unterkapitel 2.1.2 in einer Ereignis-basierten Modellierungsart noch einmal dargestellt.

2.2.1 Einführung

Bei der Ereignis-basierten Modellierung von Geschäftsprozessen stehen wie der Namen schon sagt die Ereignisse während des Prozesses im Vordergrund. Anhand der BPMN (Business Process Modeling Notation) wird hier die Ereignis-basierte Model-

lierungsart näher betrachtet. BPMN ist eine Flussdiagramm basierte Notation für die Definition von Geschäftsprozessen.

In der BPMN gibt es Flow-Objekte unter welche Ereignissen, Aktivitäten und Gateways fallen.

Ein Ereignis wird in der BPMN durch ein Kreissymbol dargestellt. Ereignisse können z. B. zeitliche Ereignisse, also das Ablaufen einer Frist, oder das Erreichen eines bestimmten Zeitpunkts sein. Ein anderes Beispiel für ein Ereignis ist das Eintreffen einer Nachricht.

Aktivitäten werden durch Rechtecke mit abgerundeten Ecken dargestellt. Im Gegensatz zu den Ereignissen handelt es sich bei Aktivitäten um konkrete Aufgaben, die ausgeführt werden.

Ein Gateway stellt in der BPMN immer einen Punkt dar, an dem der Kontrollfluss des Diagramms geteilt oder zusammengeführt wird. Es kann z. B. ein paralleler Fluss erfolgen oder eine Oder-Entscheidung getroffen werden. Dargestellt werden Gateways in der BPMN durch Rauten.

Die Connecting-Objekte beschreiben wie die einzelnen Elemente verbunden sind. Hier gibt es zum einen den Sequenzfluss, welcher die Aktivitäten, Ereignisse und Gateways verbindet und somit den Kontrollfluss des Diagramms darstellt. Zum anderen gibt es auch noch den Nachrichtenfluss, welcher den Austausch von Nachrichten zwischen verschiedenen Objekten beschreibt.

Weitere Elemente in der BPMN sind Pools und Swimlanes. Sie kennzeichnen Bereiche für unterschiedliche Benutzer und Rollen. Zwischen den einzelnen Bereichen kann ein Nachrichtenfluss stattfinden.

Neben diesen Elementen gibt es in der BPMN aber auch Artefakte. Ein Artefakt kann in der BPMN ein Datenobjekte, eine Anmerkung oder eine Gruppierung sein. Artefakte werden benutzt um zusätzliche Informationen im Diagramm darzustellen. Mit Datenobjekten können Dokumente und Datensätze modelliert werden. Ein Datenobjekt kann aber auch ein physisches Objekt sein. Der Sequenz- und Nachrichtenfluss wird von den Artefakten nicht beeinflusst. In dem Beispiel in Unterkapitel 2.2.2 wird die Nutzung von Artefakten verdeutlicht.

2.2.2 Beispiel in BPMN

Die Abbildung 2.4 zeigt die Abfertigung eines Gastes in einem Restaurant in der BPMN-Notation.

Das Modell ist in zwei Pools, den des Kunden und den des Restaurant unterteilt. Dabei ist der Restaurant Pool wiederum in die Bedienung und die Küche unterteilt. Die Aktivitäten laufen in den Pools nacheinander ab. Hier sollen besonders der Nachrichtenfluss und die Artefakte betrachtet werden.

Über den Nachrichtenfluss tauschen die Pools Informationen zwischen den Aktivitäten aus. Datenobjekte werden genutzt, um zu verdeutlichen, was genau ausgetauscht wird. Die Datenobjekte stellen hier nur zusätzliche Informationen dar und könnten auch weggelassen werden. Sie beeinflussen den Ablauf des Prozesses nicht.

Zu Beginn erhält der Kunde von der Bedienung die Speisekarten. Die Bestellung wird aufgenommen und an die Küche weiter geleitet, welche das Essen zubereitet. Nach dem Essen erhält der Kunde die Rechnung und bezahlt. Zum Schluss wird die Bilanz des Geschäfts aktualisiert.

2.3 Vergleich der Modellierungsarten

Bei der typischen Prozess Modellierung, also der Ereignis-basierten Modellierung, wird die Informationsperspektive meist ignoriert oder Informationen werden zumindest nur in Zusammenhang mit einer bestimmten Aktivität dargestellt. Dadurch kommt es, dass der Fokus von Geschäftsleuten darauf liegt, was getan werden soll, anstatt darauf, was getan werden kann [BGH⁺07]. Das führt dazu, dass Innovationen verhindert werden. Bei der Artefakt-basierten Modellierung ist dies anders, da der Blick hier auf den Artefakten und damit auf allen verfügbaren Informationen liegt. Somit können Innovationsmöglichkeiten eher entdeckt und Innovationen umgesetzt werden.

An dem Beispiel der Gast Abfertigung aus den Unterkapiteln 2.1.2 und 2.2.2 können die Unterschiede der Modellierungsansätze verdeutlicht werden.

Bei der Ereignis-basierten Modellierung werden Aktivitäten aneinander gereiht und ausgeführt. Der Kontrollfluss ist in zwei Teile unterteilt, den des Kunden und den des Restaurants. Informationen in Form von Artefakten werden nur Stellenweise angegeben und stehen nicht im gesamten Prozess-Kontext zur Verfügung.

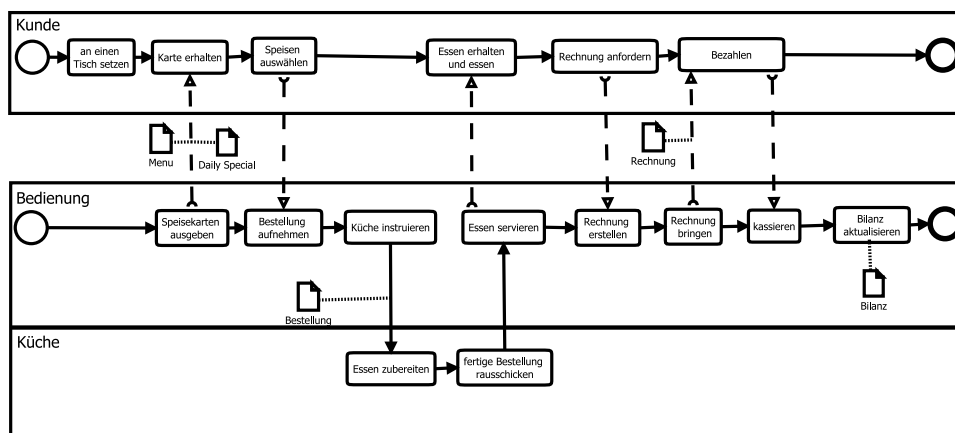


Abbildung 2.4: Beispiel Guest Check in der BPMN

Bei der Artefakt-basierten Modellierung gibt es nur wenige Services, die aktiv etwas tun. Der Kontrollfluss ist durch die Artefakte bestimmt, die zu jeder Zeit alle verfügbaren Informationen an einem Ort kapseln.

Die Artefakt-basierte Modellierung ist durch die kompakte Modellierung übersichtlicher als die Ereignis-basierte Modellierung. Allerdings ist bei der Ereignis-basierten Modellierung schneller ersichtlich, wie der Prozess abläuft. Der Prozess ist einfacher zu modellieren, da einfach danach gegangen werden kann, was in welcher Reihenfolge abläuft. Bei der Artefakt-basierten Modellierung muss zunächst ein Schlüsselartefakt bestimmt werden, dessen Lebenszyklus anschließend modelliert werden kann. Aus der Datensicht ist die Artefakt-basierte Modellierung im Vorteil. Alle Informationen werden in einem Artefakt gekapselt und stehen so jederzeit zur Verfügung. Bei der Ereignis-basierten Modellierung können die Daten nur durch zusätzliche Artefakte modelliert werden, welche den Prozessablauf allerdings nicht beeinflussen.

3 Analyse der Artefakt-basierten Modellierung

Um die Artefakt-basierte Modellierung von Geschäftsprozessen analysieren zu können, muss zunächst ein formales Modell definiert werden. In dem Paper [BGH⁺07] wird ein solches Modell definiert, um eine automatische Konstruktion von Prozessen zu ermöglichen. Anschließend werden in dem Paper durch eine statische Analyse Komplexitätsergebnisse für drei ausgewählte Probleme erarbeitet. Der Schwerpunkt des Papers liegt auf der formalen Darstellung der Artefakt-basierten Modellierung. Das formale Modell wird in Unterkapitel 3.1 genau vorgestellt. Die Analyse und deren Ergebnisse aus dem Paper werden in Unterkapitel 3.2 nur kurz erläutert.

3.1 Formales Modell

Das formale Modell der Artefakt-basierten Analyse aus dem Paper [BGH⁺07] beinhaltet Artefakte, Schemata, Services und Business-Regeln. Diese vier Elemente werden unter dem Begriff Artefaktssystem zusammengefasst. Ein Artefaktssystem hat einige nützliche semantische Eigenschaften, die bei der späteren Analyse hilfreich sind. Die Definition der einzelnen Elemente wird im folgenden anhand von Beispielen, die sich auf das zweite Beispiel aus Kapitel 2.1.2 beziehen, erläutert.

Artefakte

Um ein Artefakt formal definieren zu können werden zunächst die folgenden paarweise disjunkten, abzählbar unendlichen Mengen festgelegt. Bei den Mengen handelt es sich um die Mengen \mathcal{T}_p von primitiven Typen, \mathcal{C} von Artefaktklassennamen, \mathcal{A} von Attributnamen, $STATES$ von Artefakt-Zuständen und ID_C von Artefakt-Identifikatoren für jede Klasse $C \in \mathcal{C}$. Außerdem wird ein Typ als $\mathcal{T} = \mathcal{T}_p \cup \mathcal{C}$ festgelegt [BGH⁺07].

Die Definition einer Artefaktklasse lautet dann wie folgt.

Definition 3.1 *Eine Artefaktklasse ist ein Tupel von $(C, \mathbf{A}, \tau, Q, s, F)$.*

Dabei ist $C \in \mathcal{C}$ der Name der Klasse. $\mathbf{A} \subseteq \mathcal{A}$ ist eine endliche Menge an Attributen. Die Abbildung $\tau : \mathbf{A} \rightarrow \mathcal{T}$ ordnet den Attributen Typen zu. $Q \subseteq STATES$ ist eine endliche Menge an Zuständen. $s \in Q$ ist ein initialer und $F \subseteq Q$ eine Menge von finalen Zuständen [BGH⁺07].

Die Artefaktklasse beschreibt formal eine Klasse von Artefakten. Die Abbildung 3.1 zeigt auf der linken Seite die Artefaktklasse mit dem Namen *ORDER*. Die Artefaktklasse enthält eine Menge von Attributen, denen jeweils ein Typ zugeordnet ist.

Außerdem enthält sie eine Menge von Zuständen, bei denen der initiale Zustand und finale Zustände gekennzeichnet sind.

Die konkreten Artefakt-Objekte gehören einer solchen Artefaktklasse an und sind wie folgt definiert.

Definition 3.2 *Ein Artefakt ist ein Tripel von (o, μ, q) aus der Klasse $(C, \mathbf{A}, \tau, Q, s, F)$. Dabei ist $o \in \mathbf{ID}_C$ der Identifikator des Artefakts, μ eine Zuordnung der Attribute zu Werten und $q \in Q$ der aktuelle Zustand [BGH⁺07].*

Abbildung 3.1 stellt auf der rechten Seite ein konkretes Objekt der Artefaktklasse *ORDER* da. Das Artefakt-Objekt hat eine Identität, einen aktuellen Zustand und eine Menge an Attributen.

Nun wird noch ein Schema definiert.

Definition 3.3 *Ein Schema ist eine endliche Menge Γ von Artefaktklassen [BGH⁺07].*

Im Beispiel zwei aus Kapitel 2.1.2 besteht das Schema aus den drei Artefakt Klassen *Order*, *Tasks* und *Invoice*. Eine Instanz eines solchen Schemas wird wie folgt definiert.

Definition 3.4 *Ist Γ ein Schema, so ist eine Instanz von Γ eine Zuordnung I , welche jeder Klasse C in Γ eine endliche, valide und komplette Menge an Artefakten der Klasse C zuordnet [BGH⁺07].*

Als nächstes werden Service definiert.

Service

Um einen Service zu definieren wird zunächst die Existenz von paarweise disjunkte, abzählbar unendliche Mengen von Variablen für Klassen in \mathcal{C} vorausgesetzt. Eine Variable vom Typ $C \in \mathcal{C}$ enthält einen Identifikator in \mathbf{ID}_C . Eine Menge von Termen über ein Schema Γ hat Variablen einer Klasse C aus Γ und es gilt $x.A$, wobei x ein Term ist und A ein Attribut in C [BGH⁺07].

Ein Service ist dann durch Eingabe-Variablen, Vorbedingungen und konditionale Effekte beschrieben.

Zunächst werden Atome und Bedingungen definiert.

ARTIFACT CLASS ORDER		AN ORDER OBJECT
STATES:	ATTRIBUTES:	ID: <i>id927461</i>
PendingOrder (initial)	invoice: Invoice	STATE: <i>LiveOrder</i>
Planning	task: Task	ATTRIBUTES:
Canceled	dateCreated : String	invoice: <i>id1317231</i>
LiveOrder	crediCheckApproved : bool	task: <i>id540343</i>
Completed (final)	currentCredit: int	dateCreated: "2 April 2007"
...	installApproved: bool	crediCheckApproved: <i>true</i>
	...	currentCredit: <i>undefined</i>
		installApproved: <i>undefined</i>
		...

Abbildung 3.1: Beispiel Artefaktklasse und Artefakt [BGH⁺07]

Definition 3.5 Ein Atom ist einer der folgenden vier Ausdrücke:

- $t_1 = t_2$, wobei t_1 und t_2 Terme einer Klasse C in Γ sind,
- $DEFINED(t, A)$, wobei t ein Term ist und A ein Attribut der Klasse C ist,
- $NEW(t, A)$, wobei t ein Term und A ein Attribut ist, oder
- $s(t)$, wobei t ein Term und s ein Zustand von einer Klasse C ist [BGH⁺07].

Eine Bedingung ist dann wie folgt definiert.

Definition 3.6 Eine Bedingung ist eine Konjunktion von Atomen und negierten Atomen.

Enthält die Bedingung keine Zustandsänderung, so kann sie als Zustandslos angesehen werden. Nun wird noch ein konditionaler Effekt definiert.

Definition 3.7 Ist V eine Menge von Variablen von einer Klasse C in einem Schema Γ , dann ist ein konditionaler Effekt über V eine endliche Menge $E = \psi_1, \dots, \psi_q$ von Zustandslosen Bedingungen über V [BGH⁺07].

Nach diesen ersten Definitionen kann nun ein Service definiert werden.

Definition 3.8 Ein Service für ein Schema Γ ist ein Tupel von (n, V_r, V_w, P, E) [BGH⁺07].

Dabei entspricht $n \in \mathbb{S}$ dem Servicenamen, V_r und V_w sind endliche Mengen von Variablen von Klassen in Γ , P ist eine Zustandslose Bedingung über V ohne NEW und E ist ein konditionaler Effekt.

Abbildung 3.2 zeigt zwei Beispiel Services, den Service *UpdateCredit* und den Service *PlanSchedule*. Beide enthalten jeweils eine Menge V_r und V_w an Artefakten, auf die sie lesenden bzw. schreibenden Zugriff haben, die Vorbedingung P und konditionale Effekte E . Des Weiteren wird definiert, dass ein Service monoton ist, wenn jedes Attribut höchstens einmal geschrieben wird.

<p><u>SERVICE</u> <i>UpdateCredit</i> WRITE: {x: Order} READ: {y: CreditReport} PRE: $\neg DEFINED(x, creditCheckApproved)$ $\wedge \neg DEFINED(x, currentCredit)$ EFFECTS: – $DEFINED(x, creditCheckApproved)$ – $DEFINED(x, creditCheckApproved)$ $\wedge DEFINED(x, currentCredit)$</p>	<p><u>SERVICE</u> <i>PlanSchedule</i> WRITE: {x: Order} READ: {x: Order, s: Supplier, c: Site} PRE: $\neg DEFINED(x.task)$ EFFECTS: – $NEW(x, task) \wedge$ $DEFINED(x.task, expectedStartDate) \wedge$ $DEFINED(x.task, expectedEndDate) \wedge$ $DEFINED(x.task, supplier) \wedge$ $DEFINED(x.task, site) \wedge$ $x.task.supplier = s \wedge x.task.site = c$</p>
--	---

Abbildung 3.2: Beispiel Service [BGH⁺07]

Business-Regeln

Bevor das Artefaktsystem als ganzes beschrieben werden kann, werden noch die Business-Regeln definiert.

Definition 3.9 Bei gegebenem Schema Γ und einer Menge von Services S ist eine Business-Regel definiert, als Ausdruck der Form

”if φ invoke $\sigma(x_1, \dots, x_l; y_1, \dots, y_k)$ ”

oder

”if φ change state to ψ ,

wobei φ eine Bedingung für die Variablen $x_1, \dots, x_l, y_1, \dots, y_k$ ($l, k \geq 0$) ist und σ ein Service in S , sodass x_1, \dots, x_l alle Artefakt-Variablen sind, die modifiziert werden und y_1, \dots, y_k alle Variablen, die ausschließlich gelesen werden. ψ ist eine Bedingung, die ausschließlich aus positiven Zustandsatomen über x_1, \dots, x_l besteht [BGH⁺07].

Ein Beispiel dazu ist in Abbildung 3.3 dargestellt. Das Beispiel sagt in der ersten Zeile aus, dass der Service *InitiateAccounting* aufgerufen wird, wenn das Attribute *creditCheckApproved* des Artefakts in dem Zustand *PendingOrder* gesetzt ist. Die zweite Zeile sagt, dass eine Zustandsänderung des *Order* Artefakts zu *LiveOrder* und eine Zustandsänderung des *Tasks* Artefakts zu *PendingTask* durchgeführt wird, wenn die Attribute *expectedStartDate* und *expectedEndDate* des *Tasks* Artefakts und das Attribut *installApproved* des *Order* Artefakts definiert sind.

Artefakt System

Nachdem Schemata, Services und Business-Regeln definiert sind, kann das Artefaktsystem definiert werden.

Definition 3.10 Ein Artefaktsystem ist ein Tripel von $W = (\Gamma, S, R)$, wobei Γ ein Schema ist, S eine Menge von Services und R eine Menge von Business-Regeln [BGH⁺07].

Nun werden noch ein paar semantische Eigenschaften eines Artefaktsystems angesehen. Dazu wird zunächst ein Pfad definiert, welcher die Eigenschaften *valid*, *o-relevant*, *o-successful*, *o-dead-end* oder *o-focused* haben kann.

Definition 3.11 Ein Pfad in W ist eine endliche Sequenz von $\pi = I_0, I_1, \dots, I_n$ von Instanzen von Γ .

```

If PendingOrder(x)  $\wedge$  DEFINED(x, creditCheckApproved) invoke InitiateAccounting(x);
If DEFINED(x.task.expectedStartDate)  $\wedge$  DEFINED(x.task.expectedEndDate)
   $\wedge$  DEFINED(x, installApproved)
  change state to LiveOrder(x)  $\wedge$  PendingTask(x.task)

```

Abbildung 3.3: Beispiel Business-Regeln [BGH⁺07]

- Ein Pfad ist valid, wenn für jedes $j \in [1..n]$, I_j ein Resultat der Anwendung einer Business-Regel r aus \mathbf{R} auf I_{j-1} ist.
- o -relevant ist ein Pfad π für eine Artefakt-ID o , wenn die ID o in der Transition von I_0 nach I_1 erschaffen wird. Formal muss gelten, dass $n \geq 1$ ist und o in I_1 aber nicht in I_0 vorkommt.
- Wenn der Pfad o -relevant ist, dann ist er o -successful, wenn $I_n(o)$ ein Endzustand ist.
- Gibt es keinen validen Pfad, der zu einem o -successfulen Ende führt, so ist der Pfad o -dead-end.
- Ein valider Pfad heißt o -focused, wenn er o -relevant ist und die folgenden Eigenschaften gelten. o muss in jedem I_j , $j \in [1..n]$, vorkommen. Für jedes $j \in [2..n]$ muss gelten, dass $I_j(o) \neq I_{j-1}(o)$ ist, also entweder eine Zustandsänderung erfolgt ist oder sich ein Attribut verändert hat. Und für jedes $j \in [2..n]$ muss gelten, dass für jedes $o' \neq o$ gilt, dass $I_j(o') = I_{j-1}(o')$ ist, also keine Änderung erfolgt ist.

Mit dem Artefaktsystem $W = (\Gamma, \mathbf{S}, \mathbf{R})$ ist die Artefakt-basierte Modellierung formal beschrieben. Als nächstes wird eine beispielhafte Analyse mit Hilfe des Modells, bezogen auf drei ausgewählte Fragen, angesehen.

3.2 Analyse und Ergebnisse

Die Analyse der Artefakt-basierten Modellierung aus dem Paper [BGH⁺07] bezieht sich auf die Automatische Konstruktion von Artefaktschemata. Daher werden Komplexitäts-Charakteristika für drei Basis-Entscheidungsprobleme betrachtet. Dabei geht es um die drei Fragen: „Kann ein Artefakt erfolgreich produziert werden?“, „Existiert ein Pfad, der zu einem toten Ende im Prozess führt?“ und „Sind redundante Daten in einem Artefakt enthalten oder sind redundante Services vorhanden?“ [BGH⁺07]. Die Analyse soll dabei beispielhaft zeigen, wie mit Hilfe des formalen Modells die Artefakt-basierte Modellierung analysiert werden kann.

Formal können die Fragen wie folgt definiert werden:

Q1: Gibt es eine Artefakt-ID o und einen *validen, o -successfulen* Pfad in W dafür?

Q2: Gibt es eine Artefakt-ID o und einen *validen, o -dead-end* Pfad in W dafür? Ist das der Fall, so wird die Frage gestellt, ob es möglich ist ein Artefaktsystem W' zu erzeugen, dass äquivalent zu W ist und keinen *o -dead-end* Pfad hat.

Q3: Ist ein Attribut A von C redundant in W ?

Zunächst wird in dem Paper festgestellt, dass alle drei Probleme für ein allgemeines Artefaktsystem unentscheidbar sind. In Theorem 4.1 des Papers wird festgehalten,

dass Q1, Q2 und Q3 in PSPACE sind, wenn das Artefaktsystem W das Prädikat NEW nicht enthält [BGH⁺07].

Das Artefaktsystem wird so eingeschränkt, dass die drei Fragen NP-Vollständig sind. In Theorem 4.2 des Papers wird dafür ein monotones Artefaktsystem angenommen [BGH⁺07]. Es werden folgende drei Annahmen getroffen.

- (i) Jeder Service ist deterministisch.
- (ii) Die Vorbedingung für jeden Service ist positiv und hat kein Atom der Form $s(t)$.
- (iii) Der Vorgänger von jeder Regel ist Positiv und hat kein Atom der Form $s(t)$.

Theorem 4.3 des Papers besagt, dass Q1 für die folgenden Fälle NP-Vollständig ist [BGH⁺07].

- (a) Die Bedingungen (ii) und (iii) treffen zu, aber (i) nicht und jeder Service kann mindestens einmal ausgeführt werden.
- (b) Die Bedingungen (i), (ii) und (iii) treffen zu und Negation ist in den Vorbedingungen von Services verboten.
- (c) Die Bedingungen (i), (ii) und (iii) sind erfüllt und Negationen sind in vorangehenden Business-Regeln verboten.
- (d) In den Vorbedingungen und konditionalen Effekten werden Atome der Form $s(t)$ genutzt, aber alle anderen Bedingungen von Theorem 4.2 sind erfüllt.

Für Q2 wird die Überführung eines Artefaktsystems W in ein äquivalentes Artefaktsystem W' und die Eliminierung eines *dead-end* Pfades gezeigt. Dies ist im Paper in Theorem 4.5 festgehalten [BGH⁺07].

Zum Schluss wird gezeigt, dass die Frage Q3 unter den selben Einschränkungen wie Q1, entscheidbar ist. In Theorem 4.6 wird gesagt, dass die Entscheidung, ob ein Attribut A einer Klasse $C \in \Gamma$ redundant ist, Co-NP-Vollständig ist, für alle Fälle aus Theorem 4.3 [BGH⁺07].

4 Fazit

In dieser Ausarbeitung wurde die Artefakt-basierte Modellierung vorgestellt, bei der Geschäftsprozesse auf der Grundlage von Business Artefakten modelliert werden. In einem Business Artefakt sind alle wichtigen Informationen des Prozesses gekapselt. Der Prozess wird um ein Schlüsselartefakt modelliert, sodass alle wichtigen Informationen jederzeit vorhanden sind.

Der Vergleich zur Ereignis-basierten Modellierung aus Kapitel 2.3 ergibt, dass die Modellierung eines Geschäftsprozesses mit der Ereignis-basierten Technik auf den ersten Blick einfacher ist, da bei der Ereignis-basierten Modellierung modelliert wird, was während des Prozesses geschieht. Der Ablauf des Prozesses wird betrachtet. Es muss nicht, wie bei der Artefakt-basierten Modellierung, zuerst ein Schlüsselartefakt bestimmt werden, wonach der Prozess modelliert werden kann. Allerdings hat die Artefakt-basierte Modellierung den Vorteil, dass wichtige Informationen zu dem Prozess an einem Ort gekapselt sind und somit Innovationsmöglichkeiten besser ersichtlich sind. Bei der Ereignis-basierten Modellierung wird die Informationssicht größtenteils vernachlässigt.

In Kapitel 3 wird gezeigt, dass die Artefakt-basierte Modellierung auch als formales Modell dargestellt werden kann und dadurch Analysen des Modells möglich sind. Durch die formale Beschreibung der Artefakt-basierten Modellierung sollen auch automatisierte Konstruktionen für Artefaktschemata möglich sein. Das ist auch wichtig für die weitere Forschung.

Laut [NC03] ist die Artefakt-basierte Modellierung für Geschäftsleute leicht nachzuvollziehen und zu verstehen. Sie soll eine intuitive Art sein, um Geschäftsprozesse zu modellieren und dennoch formal Beschreibbar und Analysierbar sein. Wenn einmal ein Schlüsselartefakt gefunden ist, kann der Prozess modelliert werden. Das Problem ist dabei, dass es nicht immer einfach ist, ein solches Schlüsselartefakt zu finden.

Aus der Sicht eines Informatikers erscheint die Artefakt-basierte Methode relativ kompliziert. Für die Artefakt-basierte Modellierung darf nicht der Ablauf eines Prozesses betrachtet werden, es müssen die dabei entstehenden Informationen betrachtet werden und daraus die Modellierung erfolgen.

Der Prozessablauf bei einem Artefakt-basiert Modellierten Geschäftsprozess ist auf den ersten Blick nicht leicht ersichtlich. Artefakte können eine Service mehrfach durchlaufen und es liegt allein an den Attributen des Artefakts, was mit diesem im Weiteren Verlauf geschieht. Daraus, dass das Nachvollziehen eines Modellierten Geschäftsprozesses schon schwierig ist, kann geschlossen werden, dass die Artefakt-basierte Modellierung eines Prozesses ebenfalls nicht so einfach ist.

Die Ereignis-basierte Modellierung hingegen beschreibt den Prozessablauf und ist dadurch leicht nachzuvollziehen und zu modellieren. Die Vernachlässigung der Informationssicht ist hier natürlich ein Problem, was auf Grund der weiten Verbreitung der Ereignis-basierten Modellierung wohl eher gering einzuschätzen ist.

In Anlehnung an das Paper [Hul08] von R. Hull wird nun noch ein Blick darauf geworfen, in welche Richtung die weitere Forschung in Hinsicht auf die Artefakt-basierte Modellierung von Geschäftsprozessen gehen kann.

Eine Herausforderung ist es, ein Modell zu entwickeln, dass alle Anforderungen des Artefakt-basierten Ansatzes umsetzt. Es gibt bereits verschiedene Modelle, aber es hat sich bisher keins als das Beste herausgestellt, dass allen Anforderungen gerecht wird. Daher wird weiterhin mit verschiedenen Modellen experimentiert[Hul08].

Andere Forschungsrichtungen beschäftigt sich z. B. mit Designentscheidungen in Hinsicht auf Bedienbarkeit und Flexibilität, Implementation und Optimierung von Artefakt-basierten Ansätzen oder der Analyse von Artefakt-basierten Modellen. Es gibt also noch viel, was bei der Artefakt-basierten Modellierung betrachtet und verbessert werden kann.

Literaturverzeichnis

- [BGH⁺07] K. Bhattacharya, C. Gerede, R. Hull, R. Liu, and J. Su. Towards formal analysis of artifact-centric business process models. pages 288–304, 2007.
- [Hul08] R. Hull. Artifact-centric business process models: Brief survey of research results and challenges. pages 1152–1163, 2008.
- [NC03] A. Nigam and N.S. Caswell. Business artifacts: An approach to operational specification. *IBM Systems Journal*, 42(3):428–445, 2003.