Studienarbeit

Ein schärferes Kriterium für die Wahl von Endzuständen in Bedienungsanleitungen, Liberalsten Partnern und Public Views

Christian Gierds

22. Oktober 2007

Humboldt-Universität zu Berlin
Mathematisch-Naturwissenschaftliche Fakultät II
Institut für Informatik
Betreuer:
Peter Massuthe

Zusammenfassung

In der Welt der Service Orientierten Architektur (SOA) besteht der Bedarf, Dienste auf ihre mögliche Interaktion mit anderen Diensten hin zu untersuchen. Dienste werden wir in Form von Serviceautomaten betrachten, die als asynchron kommunizierende Automaten definiert sind. Um die Frage einer sinnvollen, also verklemmungsfreien Kommunikation zu klären, gibt es das Konzept der Bedienungsanleitungen. Wie werden für diese ein scharfes Kriterium für die Wahl der Endzustände angeben und zeigen, dass diese Wahl sich in vorhandene Konzepte integriert. Besonderes Augenmerk werden wir dabei auf den Liberalsten Partner und den Public View eines Serviceautomaten werfen und an diesen unsere Definition rechtfertigen.

Inhaltsverzeichnis

1 Einleitung **6**

2 Grundlagen **7**
2.1 Serviceautomaten und ihre Interaktion 7
2.2 Partner von Serviceautomaten und Partnergenerierung 11

3 Neues Kriterium für die Wahl der Menge der Endzustände **18**
3.1 Bisheriges Kriterium 18
3.2 Neues Kriterium 19

4 Fazit **25**

Literaturverzeichnis **26**

1 Einleitung

Wenn wir den Wunsch haben, von der Realität zu abstrahieren, werden wir zwangsläufig einen Kompromiss zwischen der Nähe zum ursprünglichen System beziehungsweise Problem und der Ausdruckskraft und Analysierbarkeit des gewählten Formalismus eingehen müssen. Dabei kann es sein, dass wir bezüglich bestimmter Aspekte eine gewisse Freiheit haben, weil diese nicht direkt auf die Realität übertragbar sind oder einfach keine Relevanz haben.

In dieser Arbeit werden wir einen intensiven Blick auf das Thema der Endzustände verschiedener, eng miteinander zusammenhängender Formalismen werfen. Die Grundlage dafür werden die Serviceautomaten bilden, die asynchron kommunizierende Dienste (zum Beispiel im Internet) repräsentieren. Während wir bei der Modellierung bis auf Randbedingungen die Wahl der Endzustände dem Modellierer überlassen, der diesen eine entsprechende Bedeutung beimisst, müssen wir Konventionen treffen, welche Zustände wir in generierten Serviceautomaten als Endzustände betrachten wollen.

Momentan werden bei der Generierung von Bedienungsanleitungen, Liberalsten Partnern und Public Views sämtliche Zustände, die nicht der Anforderung von Endzuständen in Serviceautomaten widersprechen, als Endzustand festgelegt. Dahinter steckt jedoch keine Intuition, es wird lediglich sicher gestellt, dass alle Endzustände, die relevant sein könnten, enthalten sind. Wir können aber die Menge der Endzustände einschränken, sodass sie genau die Zustände enthält, die wir benötigen, um das vorhandene Gefüge an Begriffen beizubehalten, aber keine weiteren Zustände weglassen können.

In Kapitel 2 werden wir dafür die von uns benutzten Begriffe Serviceautomat, Interaktion von Serviceautomaten, Bedienungsanleitung und Liberalster Partner einführen und motivieren. Die letzten beiden Begriffe nehmen dabei eine zentrale Rolle in dieser Arbeit ein, weil dies automatisch generierte Serviceautomaten sind, für die wir die Menge der Endzustände neu definieren wollen. Dies tun wir in Kapitel 3. Für die Begriffe Bedienungsanleitung, Liberalster Partner, oWFN des Liberalsten Partners und Public View werden wir jeweils zeigen, dass die neue Definition sinnvoll ist und dies auch anhand der Bedeutung der Begriffe motivieren. Schließlich beweisen wir, dass sich bei der Anwendung der Begriffe nichts ändert, das neue Kriterium sich also konsistent einfügt.

2 Grundlagen

Aus dem täglichen Leben und dem Umgang mit technischen Geräten haben wir eine klare Vorstellung von Bedienungsanleitungen. Diese zeigt uns in der Regel anhand von Abläufen, Schritt-für-Schritt-Anleitungen, die uns genau sagen, wie wir mit dem Gerät zu interagieren haben, damit wir ein bestimmtes Ziel erreichen. Dabei interessiert uns nicht, wie dieses Gerät funktioniert, wir möchten nur wissen, welche Knöpfe wir zum Beispiel an unserem Videorekorder drücken müssen, damit der Film am Abend aufgenommen wird.
In dieser Arbeit abstrahieren wir von dieser sehr spezifischen Art der Interaktion. Unser „Gerät“ wollen wir als *Automaten* betrachten, der uns einen bestimmten Dienst, einen *Service*, zur Verfügung stellt. Diesen nutzen wir, in dem wir Nachrichten mit dem Automaten austauschen (dies ist eine durchaus sinnvolle Abstraktion; das Drücken von Knöpfen oder Lesen von Statusmeldungen auf einem Display kann man als Nachrichtenaustausch betrachten).
Wir werden zum einen den Begriff des *(Service–)Automaten* formalisieren, zum anderen werden wir darauf aufbauend den Begriff der *Bedienungsanleitung* herleiten, mit dem wir jede sinnvolle Interaktion mit einem gegebenen Automaten charakterisieren können.

Die Definitionen in den folgenden beiden Abschnitten sind bereits etabliert und sind in dieser Form [Lohmann et al., 2007b] entnommen. Um eine möglichst große Kohärenz mit bestehender Literatur sicherzustellen, benutzen wir für bestimmte Bezeichnungen die englischen Begriffe.

2.1 Serviceautomaten und ihre Interaktion

Wir wollen ein Objekt beschreiben, von dem uns vorrangig der Nachrichtenaustausch mit der Umgebung (einem Partner) interessiert – ein zustandsbasiertes System, bei dem die Übergänge zusammen mit einem Nachrichtenaustausch stattfinden.

Dafür wollen wir den Begriff des *Serviceautomaten* definieren. Dieser ist elementar und alle weiteren Definitionen bauen auf ihm auf.

Definition 1 (Serviceautomat)
Ein Serviceautomat *ist ein nicht-deterministischer Automat* $A = [Q, I, O, \delta, q_0, F]$ *mit*

- *einer endlichen Menge* Q *von Zuständen,*
- *einem Interface* $I \cup O$*, mit Eingabekanälen* I *(input) und Ausgabekanälen* O *(output)* $(I \cap O = \emptyset)$*,*
- *einer Übergangsrelation* $\delta \subseteq Q \times (I \cup O \cup \{\tau\}) \times Q$*,*
- *einem Anfangszustand* $q_0 \in Q$ *sowie*
- *einer Menge von Endzustände* $F \subseteq Q$*, wobei für alle* $q \in F$ *gilt, dass* $x \in I$ *aus* $[q, x, q'] \in \delta$ *folgt.*

⌟

Die Grundlage für den Serviceautomaten bildet ein Transitionssystem, bei dem die Zustandsübergänge beschriftet sind. Diese Übergänge $[q, x, q'] \in \delta$ bestehen entweder aus dem Senden ($x \in O$) oder Empfangen ($x \in I$) einer Nachricht, bzw. einem internen Übergang ($x = \tau$), der für die Umgebung nicht sichtbar ist. In der graphischen Notation und auch im weiteren Verlauf dieser Arbeit werden wir die Schreibweise $?x$ benutzen, um zu kennzeichnen, dass x empfangen wird, $!x$, wenn x gesendet wird, bzw. τ im Falle eines internen Übergangs.

Zustände, die der Serviceautomat nicht selbständig (durch Senden einer Nachricht oder einen internen Übergang) verlassen kann, möchten wir als *Wartezustände* bezeichnen.

Definition 2 (Wartezustand)
Sei A *ein Serviceautomat. Dann heißt ein Zustand* $q \in Q$ *genau dann* Wartezustand*, wenn* $x \in I$ *für alle* $[q, x, q'] \in \delta$ *gilt. Die Abbildung wait sei für Wartezustände wie folgt definiert:* $wait(q) = \{x \in I \mid \exists q' \in Q : [q, x, q'] \in \delta\}$*.* ⌟

Somit sind Endzustände insbesondere Wartezustände.

Wartezustände, die keine Endzustände sind, und in denen auch das Empfangen nicht mehr möglich ist, möchten wir als *Verklemmungszustände* bezeichnen. Der Serviceautomat stoppt in diesen Zuständen die Abarbeitung, obwohl kein definiertes Ende erreicht wurde.

Definition 3 (Verklemmung)
Sei A ein Serviceautomat und $q \in Q$ ein Wartezustand von A. Dann bezeichnen wir q genau dann als Verklemmungszustand, *wenn $q \notin F$ und $wait(q) = \emptyset$.* ⌟

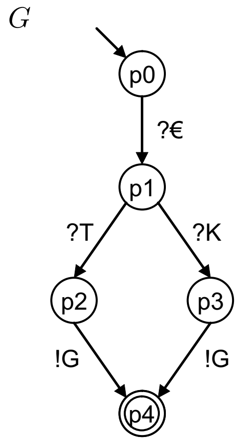

Abbildung 2.1: Der Serviceautomat G soll einen Automaten für warme Getränke modellieren. Dieser Getränkeautomat gibt entweder Tee oder Kaffee heraus, wenn ein Nutzer entsprechend Geld einwirft und danach den Knopf für Tee oder Kaffee drückt.
Im Anfangszustand p0 (gekennzeichnet durch eingehenden Pfeil) kann der Serviceautomat Geld empfangen (?€) und wechselt in den Zustand p1. Im Zustand p1 kann der Serviceautomat durch Empfangen der Nachricht für Tee (?T) in den Zustand p2 bzw. der Nachricht für Kaffee (?K) in den Zustand p3 übergehen. In den beiden Zuständen p2 und p3 geht der Automat durch Senden der Nachricht für die Ausgabe des Getränkes (!G) in den Endzustand p4 (erkennbar an der doppelten Linie) über.
Die Zustände p0, p1 und p4 sind Wartezustände. Verklemmungszustände gibt es in diesem Serviceautomaten nicht.

Unter der Kommunikation mit der Umgebung können wir im Besonderem die Kommunikation mit einem anderen Serviceautomaten verstehen. Dabei muss gewährleistet sein, dass die Nachrichten, die ein Serviceautomat sendet, der andere empfangen kann.

Definition 4 (Interface-kompatible Automaten)
Zwei Serviceautomaten A und B heißen genau dann kompatibel bezüglich ihres Interfaces, *wenn $I_A = O_B$ und $O_A = I_B$ gilt.* ⌟

Wir kommen auf den Getränkeautomaten G zurück, der das Interface $I = \{€, \mathsf{T}, \mathsf{K}\}$ und $O = \{\mathsf{G}\}$ besitzt. In Abbildung 2.2 zeigen wir zwei zu G Interface-kompatible Automaten T und K mit dem entsprechenden Interface $I = \{\mathsf{G}\}$ und $O = \{€, \mathsf{T}, \mathsf{K}\}$. Der Automat T hat keinen Übergang K und der Automat K besitzt keinen T-Übergang; dies ist jedoch kein Widerspruch zu Definition 1, da wir dort nicht fordern, dass alle Kommunikationskanäle genutzt werden müssen, die definiert werden.

Ein wichtiger Aspekt der Kommunikation in unserem Modell ist der asynchrone Nachrichtenaustausch. Dies bedeutet, dass ein Automat nicht blockiert, bis eine von ihm gesendete Nachricht empfangen wurde, und auch, dass sich Nachrichten gegenseitig überholen können. Neben den Zuständen, in denen sich die Automaten befinden können, müssen wir uns folglich auch merken, welche Nachrichten gesendet, aber noch nicht empfangen wurden. Hierfür bietet sich die Nutzung einer Multimenge aus dem System aller Multimengen $bags(C)$ über die Menge aller (Ein- und Ausgabe-)Kanäle $C = I \cup O$ an,

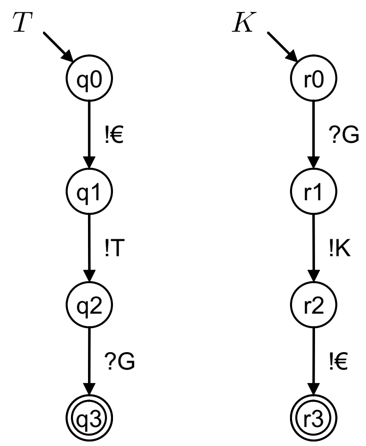

Abbildung 2.2: Die Abbildung zeigt die beiden zu G (Abb. 2.1) *Interface-kompatiblen* Serviceautomaten T und K. Den Serviceautomaten T könnte man als Kunden, der einen Tee trinken möchte, interpretieren. Entsprechend wird vom Startzustand q0 die Nachricht € (Geld einwerfen) gesendet, die Wahl für Tee getroffen (!T) und anschließend das Getränk erwartet (?G). Entsprechend könnte man den Serviceautomaten K als einen Kaffeetrinker ansehen. Dieser erwartet im Startzustand r0 bereits das Getränk (?G), um erst danach das Geld zu bezahlen (!€) und die Wahl für Kaffee zu treffen (!K).

da Nachrichten mehrfach gesendet werden können, die Reihenfolge der Nachrichten aber uninteressant ist.

Definition 5 (Komponierter Serviceautomat)
Seien A und B zwei Interface-kompatible Automaten. Die Komposition *von A und B ist definiert als Serviceautomat $A \oplus B = [Q_{A\oplus B}, I_{A\oplus B}, O_{A\oplus B}, \delta_{A\oplus B}, q_{0_{A\oplus B}}, F_{A\oplus B}]$ mit*

- $Q_{A\oplus B} = Q_A \times Q_B \times bags(C)$,
- $I_{A\oplus B} = O_{A\oplus B} = \emptyset$,
- $q_{0_{A\oplus B}} = [q_{0_A}, q_{0_B}, []]$ *und*
- $F_{A\oplus B} = F_A \times F_B \times \{[]\}$,

wobei die Übergangsrelation die Elemente

- $[[q_A, q_B, m], \tau, [q'_A, q_B, m]]$ *gdw.* $[q_A, \tau, q'_A] \in \delta_A$ *(interner Übergang in A),*
- $[[q_A, q_B, m], \tau, [q_A, q'_B, m]]$ *gdw.* $[q_B, \tau, q'_B] \in \delta_B$ *(interner Übergang in B),*
- $[[q_A, q_B, m + [x]], \tau, [q'_A, q_B, m]]$ *gdw.* $[q_A, x, q'_A] \in \delta_A$, $x \in I_A$ *(A empfängt x),*

- $[[q_A, q_B, m + [x]], \tau, [q_A, q'_B, m]]$ *gdw.* $[q_B, x, q'_B] \in \delta_B$, $x \in I_B$ *(B empfängt x),*
- $[[q_A, q_B, m], \tau, [q'_A, q_B, m + [x]]]$ *gdw.* $[q_A, x, q'_A] \in \delta_A$, $x \in O_A$ *(A sendet x),*
- $[[q_A, q_B, m], \tau, [q_A, q'_B, m + [x]]]$ *gdw.* $[q_B, x, q'_B] \in \delta_B$, $x \in O_B$ *(B sendet x)*

und keine weiteren enthalten soll. ⌟

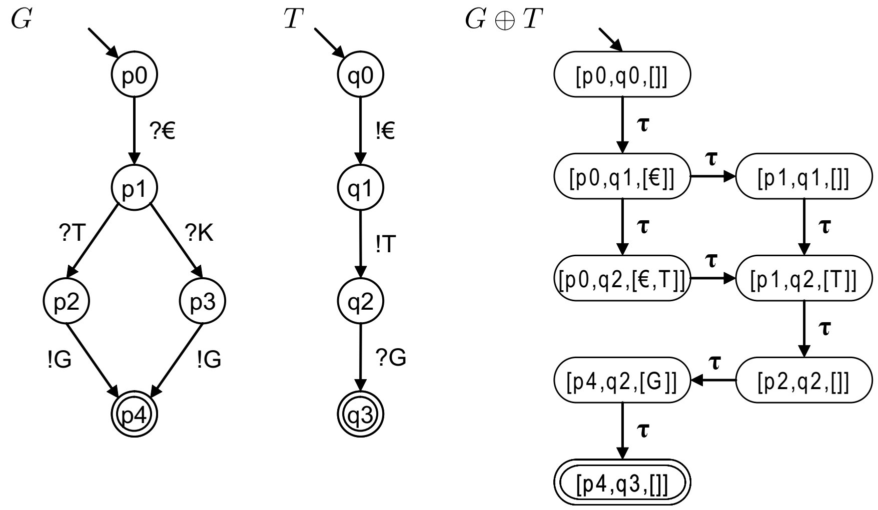

Abbildung 2.3: Die Abbildung zeigt die Komposition $G \oplus T$ der Serviceautomaten G und T. Die Tripel in den Zuständen geben an, in welchem Zustand sich G und T befinden, sowie welche Nachrichten von einem der Serviceautomaten gesendet, vom anderen jedoch noch nicht empfangen wurden. Wir sehen, dass sämtlich Abläufe im Zustand [p4, q3, []] enden, der Endzustand ist.
Der komponierte Serviceautomat aus Getränke- und Kaffeetrinker-Automaten $G \oplus K$ besteht nur aus dem Anfangszustand, der gleichzeitig auch ein Verklemmungszustand ist (weshalb er hier auch nicht gezeigt wird). Er hat keine Folgezustände, da die Partner beide auf eine Nachricht warten, also keiner voran schreiten kann.

2.2 Partner von Serviceautomaten und Partnergenerierung

Nachdem wir im vorhergehenden Abschnitt Serviceautomaten eingeführt und ihre Kommunikation miteinander beschrieben haben, steht in diesem Abschnitt die Frage im Vordergrund, wie wir zu einem gegebenen Serviceautomaten A einen Partner B generieren können, mit dem A verklemmungsfrei kommunizieren kann. Im speziellen wollen wir einen annotierten Partner, die Bedienungsanleitung, entwickeln, die alle möglichen Partner eines Serviceautomaten charakterisiert.

Definition 6 (Partner)
Seien A und B zwei Interface-kompatible Serviceautomaten. Dann bezeichnen wir B genau dann als Partner *von A, wenn vom Anfangszustand $q_{0_{A\oplus B}}$ des komponierten Serviceautomaten $A \oplus B$ kein Verklemmungszustand erreichbar ist.* ⌟

Partner können wir gemäß [Massuthe and Wolf, 2006] folgendermaßen beschreiben:

Lemma 2.1 (Charakterisierung von Partnern)
Seien A und B zwei Interface-kompatible Automaten. Dann ist B genau dann ein Partner *für A, wenn für alle $q_B \in Q_B$ und alle $[q_A, M] \in K_{B,A}(q_B)$ gilt:*

- *$[q_A, q_B, M]$ ist ein Endzustand von $A \oplus B$; oder*
- *es existiert ein Übergang $[q_A, x, q'_A] \in \delta_A$ und ein Übergang von $[q_A, q_B, M]$ in $A \oplus B$, der mit x beschriftet ist; oder*
- *es existiert ein Übergang $[q_B, x, q'_B] \in \delta_B$ und ein Übergang von $[q_A, q_B, M]$ in $A \oplus B$, der mit x beschriftet ist.*

⌟

Dieses Lemma folgt direkt aus der Definition für *Partner*.

Die Definition eines Partner ist symmetrisch, das heißt, wenn B ein Partner von A ist, so ist A auch ein Partner von B. Im Folgenden wollen wir diese Symmetrie aufbrechen: Wir werden uns auf einen gegebenen Serviceautomaten A konzentrieren und erklären, welches Wissen ein potentieller Partner über A haben könnte, um daraus einen speziellen Partner von A generieren können.

Dabei wird uns interessieren, in welchen Zuständen sich der Automat A befinden kann und welche Nachrichten unterwegs sein können, wenn sich B in einem bestimmten Zustand befindet. Diese Zuordnung eines Zustandes von B auf Situationen von A, also einen Zustand und einer Menge von Nachrichten, wollen wir als Wissen bezeichnen.

Definition 7 (Situation)
Sei A ein Serviceautomat, q ein Zustand von A und M eine Multimenge von Nachrichten über dem Interface von A. Dann bezeichnen wir das Tupel $[q, M]$ als Situation.

⌟

Definition 8 (Wissensfunktion K)
Seien A und B zwei Interface-kompatible Serviceautomaten. Dann sei die Wissensfunktion *(Knowledge) $K_{(B,A)} : Q_B \to \mathcal{P}(Q_A \times bags(C))$, die einem Zustand von B eine Menge von Situationen zuordnet, definiert durch die Abbildung $K_{(B,A)}(q_B) = \{[q_A, M] \mid [q_A, q_B, M]$ ist erreichbar von $q_{0_{A\oplus B}} \in Q_{A\oplus B}\}$.* ⌟

Die Funktion $K_{(B,A)}$ beschreibt das Wissen von B über A. In Abbildung 2.4 zeigen wir ein Beispiel für das Wissen in einem ausgewähltem Zustand.

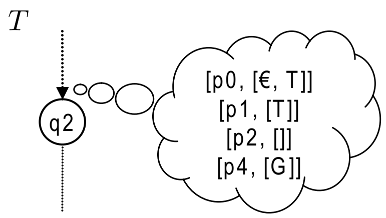

Abbildung 2.4: In $G \oplus T$ (Abbildung 2.3) ist zu sehen, dass die Zustände, in denen T im Zustand q2 ist, [p0, q2, [€, T]], [p1, q2, [T]], [p2, q2, []] und [p4, q2, [G]] sind. Daraus ergibt sich $K_{(T,G)}(\mathsf{q2}) = \{$[p0, [€, T]], [p1, [T]], [p2, []], [p4, [G]]$\}$. Die Elemente von $K_{(T,G)}(\mathsf{q2})$ sind somit die Situationen, die q2 zugeordnet sind.

Die Situationen haben ein für uns wichtiges Unterscheidungsmerkmal. Wenn wir eine Menge M von Situation des Automaten A betrachten, gibt es darunter solche, für die ein Übergang in A existiert, so dass die resultierende Situation wieder in M liegt. Diese Situation wollen wir als *transient* bezeichnen. Wichtiger für uns werden jedoch die nicht-transienten, die *stabilen* Situationen sein.

Definition 9 (Transiente und stabile Situationen)
Sei $\mathfrak{M}$ eine Menge von Situationen. Dann heißt $[q_A, M] \in \mathfrak{M}$ genau dann transient *in $\mathfrak{M}$, wenn ein $[q_A, x, q'_A] \in \delta_A$ existiert, so dass*

- *$x = \tau$ und $[q'_A, M] \in \mathfrak{M}$ oder*
- *$x \in I_A$, $M(x) > 0$ und $[q'_A, M - [x]] \in \mathfrak{M}$ oder*
- *$x \in O_A$ und $[q'_A, M + [x]] \in \mathfrak{M}$.*

Anderenfalls heißt $[q_A, M]$ stabil *in $\mathfrak{M}$.* ⌟

Haben wir eine Menge von Situationen gegeben wird es uns später interessieren, welche Situationen wir erhalten, wenn der Automat durch Eintreten eines *Ereignisses* einen Übergang machen kann.

Definition 10 (Ereignis)
Sei $M \subseteq Q_A \times bags(C)$ eine Menge von Situationen. Dann erhalten wir durch Eintreten eines Ereignisses *(event) eine neue Menge von Situationen:*

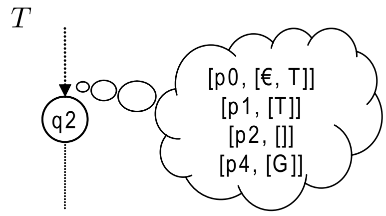

Abbildung 2.5: In Abbildung 2.4 haben wir das Wissen $K_{(T,G)}(\mathsf{q2})$ für den Zustand $\mathsf{q2} \in Q_T$ berechnet. Die Situation $[\mathsf{p0}, [€, \mathsf{T}]]$ ist transient, da der Automat G durch Empfangen der Nachricht € in einen Zustand übergehen kann, der als Situation ($[\mathsf{p1}, [\mathsf{T}]]$) in der Menge $K_{(T,G)}(\mathsf{q2})$ enthalten ist. Gleiches gilt auch für die Situationen $[\mathsf{p1}, [\mathsf{T}]]$ und $[\mathsf{p2}, []]$. Die Situation $[\mathsf{p4}, [\mathsf{G}]]$ jedoch ist stabil, da der Automat G in keinen Zustand übergehen kann, so dass eine entsprechende Situation in $K_{(T,G)}(\mathsf{q2})$ enthalten ist.

$$event(M, x) = \begin{cases} \{[q, m + [x]] \mid [q, m] \in M\}, & \textit{falls } x \in O_A \textit{ (Sendeereignis)} \\ \{[q, m] \mid [q, m + [x]] \in M\}, & \textit{falls } x \in I_A \textit{ (Empfangsereignis)} \\ M, & \textit{falls } x = \tau \textit{ (Internes Ereignis)} \end{cases}$$

⌟

Wenn wir einen Serviceautomaten A und zu diesem eine Menge $\mathfrak{M}$ von Situationen gegeben haben, ist es möglich, dass es einen Übergang in A gibt, so dass wir eine weitere Situation erreichen können, die nicht in $\mathfrak{M}$ enthalten ist. Dies entspricht der Suche nach Situationen, die durch A von einer gegeben Situation aus ohne Aktion durch einen zweiten Automaten erreicht werden können. Das Bestimmen dieser Situationen wollen wir als *Abschluss* bezeichnen.

Definition 11 (Abschluss)
Sei $\mathfrak{M}$ eine Menge von Situationen. Dann ist der Abschluss $cl(\mathfrak{M})$ *(closure) von $\mathfrak{M}$ induktiv definiert:*
Anfang: $\mathfrak{M} \subseteq cl(\mathfrak{M})$
Schritt: Wenn $[q_A, M] \in cl(\mathfrak{M})$ *und* $[q_A, x, q'_A] \in \delta_A$*, dann*

- $[q'_A, M] \in cl(\mathfrak{M})$*, wenn* $x = \tau$*,*
- $[q'_A, M + [x]] \in cl(\mathfrak{M})$*, wenn* $x \in O_A$*,*
- $[q'_A, M - [x]] \in cl(\mathfrak{M})$*, wenn* $x \in I_A$ *und* $M(x) > 0$*.*

⌟

Anhand des Abbildungen 2.5 und 2.3 können wir sehen, dass zum Beispiel der Abschluss von $K_{(T,G)}(\mathsf{q2})$ die Menge selbst ist ($cl(K_{(T,G)}(\mathsf{q2})) = K_{(T,G)}(\mathsf{q2})$). Der Abschluss der Menge $\{[\mathsf{p1}, [\mathsf{T}]]\}$ ist $cl(\{[\mathsf{p1}, [\mathsf{T}]]\}) = \{[\mathsf{p1}, [\mathsf{T}]], [\mathsf{p2}, []], [\mathsf{p4}, [\mathsf{G}]]\}$.

In diesem Abschnitt wollen wir letztendlich einen Partner B für einen gegebenen Serviceautomaten A bestimmen. Die Grundlagen dafür haben wir bereits. Mit Hilfe der Definitionen 10 (Ereignis) und 11 (Abschluss) können wir einen Serviceautomaten erzeugen, der mit A interagieren kann. Jedoch besteht die Möglichkeit, dass die Komposition der beiden Serviceautomaten in Zustände gelangt, die gegen Lemma 2.1 verstoßen, sprich in einen Verklemmungszustand, womit der erzeugte Serviceautomat kein Partner von A wäre. Um dem Rechnung zu tragen, wollen wir den erzeugten Serviceautomaten gemäß Lemma 2.1 annotieren.

Definition 12 (Annotation, Belegung)
Seien A und B zwei Interface-kompatible Serviceautomaten. Dann sei für jedes $q_B \in Q_B$ die Annotation $\phi(q_B)$ *folgendermaßen als aussagenlogische Formel über die Literale $C \cup \{\tau, \mathit{final}\}$ (mit $\tau, \mathit{final} \notin C$) definiert:*

$$\phi(q_B) := \bigwedge_{[q_A,m] \text{ ist stabil in } K(q_A)} (\phi_1(q_A, m) \vee \phi_2 \vee \phi_3(m))$$

wobei

- $\phi_1(q_A, m) := \begin{cases} \mathit{final}, & \text{falls } q_A \in F_A \text{ und } m = [] \\ \mathit{false}, & \text{sonst} \end{cases}$,
- $\phi_2 := \tau \vee \bigvee_{x \in O_B} x$,
- $\phi_3(m) := \bigvee_{x \in I_B, m(x) > 0} x$.

Die Belegung *(assignment) $ass_B(q_B) : C \cup \{\tau, \mathit{final}\} \rightarrow \{\mathit{true}, \mathit{false}\}$ ordnet den Literalen $x \in C \cup \{\tau\}$ genau dann den Wert true zu, wenn ein q'_B existiert, so dass $[q_B, x, q'_B] \in \delta_B$, und dem Literal final genau dann den Wert true, wenn $q_B \in F_B$.* ⌟

Die Annotationen spiegeln genau die drei Bedingungen aus Lemma 2.1 wider.

Mit den vorangegangen Definitionen ist es uns nun möglich, die *Bedienungsanleitung* eines Serviceautomaten A zu bestimmen. Diese wird dann alle Partner von A beschreiben.

Definition 13 (Bedienungsanleitung)
Sei A ein Serviceautomat. Dann definieren wir induktiv eine Folge von Serviceautomaten $S_i = [Q_i, I_i, O_i, \delta_i, q_{0_i}, F_i]$:
Sei $Q_0 = \mathcal{P}(Q_A \times bags_k(C))$ und für alle i sei

- $I_i = O_A$,
- $O_i = I_A$,
- $q_{0_i} = cl(\{[q_{0_A}, []]\})$,
- $[q, x, q'] \in \delta_i$, *gdw.* $q, q' \in Q_i$ *und* $[q', []] \in cl(event(\{[q, []]\}, x))$ *und*
- $F_i = \{q \in Q_i \mid q$ *ist Wartezustand von* $S_i\}$.

Dann sei $Q_{i+1} = \{q \mid q \in Q_i, \phi(q)$ *ist wahr für die Belegung* $ass_{S_i}(q)\}$.
Als Bedienungsanleitung *(OG, Operating Guideline) bezeichnen wir den Serviceautomaten mit dem kleinsten i, für den* $S_i = S_{i+1}$ *gilt (*$OG = S_i$*).* ⌟

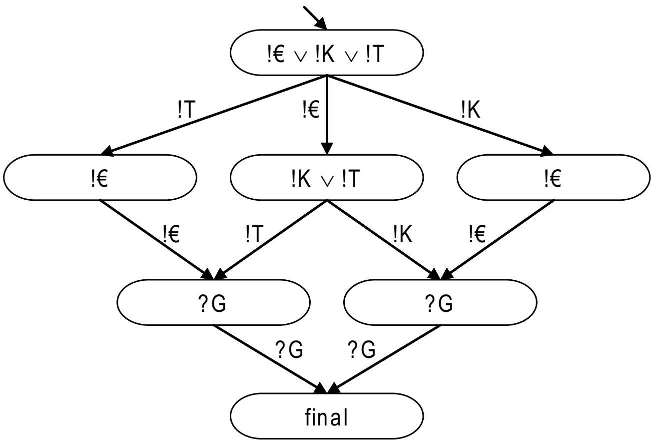

Abbildung 2.6: Hier sehen wir ein Beispiel für eine Bedienungsanleitung, wie sie mit dem Werkzeug `Fiona` ([Lohmann et al., 2007a]) berechnet werden kann.

Wir haben behauptet, dass die Bedienungsanleitung alle Partner eines gegeben Serviceautomaten repräsentiert. Die Menge aller Partner eines Serviceautomaten anzugeben ist in der Regel unmöglich, da sie meist unendlich ist. Wir können jedoch ein Matching zwischen einem Partner und der Bedienungsanleitung angeben, das heißt, ein Serviceautomat B ist genau dann ein Partner des Serviceautomaten A, wenn es ein Matching von B mit der Bedienungsanleitung von A gibt.

Definition 14 (Matching)
Seien A und B zwei Serviceautomaten. Dann gibt es genau dann ein Matching *von B mit der Bedienungsanleitung* OG_A *von A, wenn* OG_A *den Serviceautomaten B simuliert, und die Knoten von B die Annotationen der in Simulation stehenden Knoten von* OG_A

erfüllen.

⌟

Die Bedeutung des Matching liefert uns folgendes Lemma.

Lemma 2.2 (Matching)
Seien A und B zwei Serviceautomaten. Dann ist B genau dann ein Partner von A, wenn es ein Matching von B mit der Bedienungsanleitung von A gibt. ⌟

Der Beweis dieser Aussage folgt sofort aus den gegebenen Definitionen und vor allem aus Lemma 2.1.

Mit Hilfe von Definition 14 können wir somit die Menge aller Partner durch die Bedienungsanleitung charakterisieren.

Als Liberalsten Partner wollen wir nun den Serviceautomaten verstehen, welcher das größtmögliche Verhalten bezüglich eines gegebenen Automaten A aufweist, der also jeden möglichen sinnvollen Ablauf eines anderen Partners von A bewerkstelligen kann. Dies leistet gerade die Struktur der Bedienungsanleitung. Wenn wir bei dieser die Annotationen weglassen, erhalten wir einen Serviceautomaten, der offensichtlich Definition 14 erfüllt.

Definition 15 (Liberalster Partner)
Sei A ein Serviceautomat und OG die dazugehörige Bedienungsanleitung. Dann bezeichnen wir $L = (OG$ unter Weglassung der Annotation$)$ als Liberalsten Partner *von A.*

⌟

3 Neues Kriterium für die Wahl der Menge der Endzustände

Im vorhergehenden Kapitel haben wir alle für uns notwendigen Begriffe eingeführt. Auf diesen aufbauend wollen wir nun zum eigentlichen Gegenstand dieser Arbeit kommen, den Endzuständen der Bedienungsanleitung, des Liberalsten Partners und des dazugehörigen Petrinetzes, sowie des Public Views. Dazu werden wir den momentanen Stand der Definitionen betrachten, mögliche Änderungen an diesen erörtern, schließlich eine schärfere Fassung für die Menge der Endzustände bestimmen und sie in Bezug auf die bestehende Begriffswelt rechtfertigen.

3.1 Bisheriges Kriterium

Als wir den Begriff Serviceautomat in Definition 1 eingeführt haben, war die einzige Anforderung an die Endzustände, dass sie Wartezustände sein sollten. Diese Festlegung ist sinnvoll; von einem Zustand, den der Automat selbständig durch ein internes oder Sendeereignis verlassen kann, wollen wir annehmen, dass dieser auch verlassen wird. Die genaue Wahl der Endzustände wird dem Modellierer überlassen.

Bei der Generierung der Bedienungsanleitung (Definition 13) wird festgelegt, dass jeder Wartezustand in dem erzeugten Serviceautomaten Endzustand sein sollen. Dieses entspricht den Definitionen, ist jedoch etwas willkürlich, da, wie wir zeigen werden, eine in der Regel kleinere Menge von Endzuständen ausreichend ist, die zudem ein noch stärkeren Bezug zum ursprünglich gegebenen Serviceautomaten hat.

Während wir bei den Bedienungsanleitung und ihrer Anwendung sehen werden, dass die Menge der Endzustände irrelevant ist, widerspricht die ursprüngliche Definition beim Liberalsten Partner sogar unserer Intuition. Wir würden erwarten, dass der Liberalste Partner eines gegebenen Serviceautomaten höchstens dann in einem Endzustand ist, wenn der Serviceautomat selbst dies auch ist. Da der Liberalste Partner spezifisch für einen Serviceautomaten ermittelt wird, ergeben Endzustände, die in der Komposition von Serviceautomat und Liberalstem Partner nicht zu einem gemeinsamen Endzustand führen, keinen Sinn – der Liberalste Partner wird diese Zustände zwangsläufig verlassen. Aus diesem Grund werden wir solche Zustände in der neuen Definition nicht mehr berücksichtigen. Beim Public View haben die Endzustände eine noch größere Bedeutung, da dieser verhaltensäquivalent zum ursprünglichen Serviceautomaten sein soll. Insbesondere sollte er nicht in mehr Situationen in einem Endzustand sein, als es vom Modellierer gedacht war.

3.2 Neues Kriterium

Anhand dieser Betrachtung wollen wir folgende Anforderungen an unsere Endzustände stellen: Sie müssen Wartezustände sein, und der ursprüngliche Serviceautomat soll sich auch in einem Endzustand befinden können.

Definition 16 (Endzustände in der Bedienungsanleitung OG)
Sei A ein Serviceautomat und OG dessen Bedienungsanleitung. Dann definieren wir die Endzustände F_{OG} *wie folgt:*
$F_{OG} := \{q \in Q_{OG} \mid q \textit{ ist Wartezustand von } OG \textit{ und } \phi(q) \textit{ enthält die Klausel final}\}$ ⌟

Die Definition leistet genau das Geforderte. Die Menge besteht nur aus Wartezuständen und die Klausel *final* besagt, dass der Serviceautomat in einem Endzustand sein kann.

Klauseln der Form $\mathit{final} \vee a \vee \ldots$ wollen wir explizit ausschließen, da sie laut Definition 12 bedeuten, dass noch Nachrichten unterwegs sind, das heißt, dass wir keinen gültigen Endzustand erreicht haben.

In den folgenden Abschnitten werden für die Bedienungsanleitungen, die Liberalsten Partner und die Public Views zeigen, dass die neue Definition nicht nur sinnvoll ist, sondern dass wir auch keine kleinere Menge an Endzuständen finden können – das Kriterium ist somit notwendig und hinreichend.

3.2.1 Endzustände in Bedienungsanleitungen

Wir haben bereits erwähnt, dass eine Bedienungsanleitung OG_A die Menge aller möglichen Partner eines Serviceautomaten charakterisiert. Dementsprechend besteht direkte Anwendung einer Bedienungsanleitung in der Suche nach einem Matching mit einem potentiellen Partner B gemäß Definition 14. Dabei wird jedoch keine Aussage über die Endzustände von OG_A getroffen. Ein Zustand in einem Partner kann sogar Endzustand sein, obwohl der simulare Zustand in der Bedienungsanleitung keiner ist, weil im Partner im Vergleich zur Bedienungsanleitung Übergänge fehlen können, und somit andere Zustände zu Wartezuständen werden können.

Hier sind die Annotationen wichtig. Ein Zustand in einem Partner kann höchstens dann Endzustand sein, wenn er die Klausel *final* erfüllt. Insofern ist es intuitiv, dass wir in unserer Definition nur solche Zustände in die Menge der Endzustände mitaufnehmen.

3.2.2 Endzustände im Liberalsten Partner und dem entsprechende oWFN

Der Liberalste Partner L ist laut Definition 15 die Struktur der Bedienungsanleitung, also ohne Annotation, und stellt somit einen speziellen Partner eines gegebenen Serviceautomaten A dar. In diesem Sinne erwarten wir auch, dass die Endzustände im Bezug zum Serviceautomaten stehen. Das soll heißen, dass ein Zustand des Liberalste Partner kein Endzustand zu sein braucht, wenn dieser Zustand in keinem Endzustand des komponierten Systems $A \oplus L$ enthalten ist.

Lemma 3.1 (Endzustände im Liberalsten Partner)
Seien A ein Serviceautomat und L dessen Liberalster Partner. Dann ist $q_L \in Q_L$ genau dann ein Endzustand von L, wenn es einen Endzustand $q_A \in F_A$ in A gibt, sodass der Zustand $[q_A, q_L, []]$ Endzustand von $A \oplus L$ ist. ⌟

Beweis.

- „→:“ Sei $q_L \in Q_L$ ein Endzustand von L. Laut unserer Definition der Endzustände (Definition 16) hat somit die Annotation von q_L die Klausel *final* enthalten. Nach Definition 12 (Annotation) gibt es dann eine Situation $[q_A, []] \in K_{L,A}(q_L)$, wobei q_A Endzustand von A ist. Und entsprechend ist $[q_A, q_L, []]$ Endzustand von $A \oplus L$.
- „←:“ Sei $q_A \in F_A$ ein Endzustand von A, $q_L \in Q_L$ ein Wartezustand von L und $[q_A, q_L, []]$ ein Zustand von $A \oplus L$. Damit ist $[q_A, []]$ eine stabile Situation im Wissen $K_{L,A}(q_L)$ von q_L, weshalb q_L mit der Klausel *final* annotiert ist (Definition 12). Folglich ist q_L ein Endzustand von L (neues Kriterium, Definition 16).

□

Anhand des Lemmas sehen wir, dass wir für den Liberalsten Partner L eines Serviceautomaten eine genaue Charakterisierung der Menge der Endzustände erhalten haben. Weniger Zustände dürfen es nicht sein, da sonst die Annotation der Bedienungsanleitung verletzt wäre und somit L kein Partner wäre, mehr brauchen es nicht zu sein, weil alle weiteren Zustände von L im komponierten System nicht zu Endzuständen beitragen.

Serviceautomaten sind der Formalismus unserer Wahl, um die vorgestellten Begriffe zu definieren. Jedoch heißt dies nicht, dass wir unbedingt direkt mit Serviceautomaten modellieren. Ein möglicher Ausgangspunkt sind offene Workflownetze (open Workflow Nets, oWFN), für die es eine Übersetzung in Serviceautomaten gibt.

Definition 17 (open Workflow Net)
Ein open Workflow Net (oWFN) *ist ein Petrinetz zusammen mit Platzmengen P_I und P_O sowie einer Menge von Endmarkierungen Ω, so dass*

- *die Mengen der Eingabeplätze $P_I \subseteq P$ (nur ausgehende Kanten) und der Ausgabeplätze $P_O \subseteq P$ (nur eingehende Kanten) disjunkt ($P_I \cap P_O = \emptyset$) sind,*
- *und für die Menge der Endmarkierungen Ω gilt, dass $m \in \Omega$ und $p \in P_I \cup P_O$ $m(p) = 0$ bedeutet, und dass durch $m \in \Omega$ keine Transition aktiviert wird.*

⌟

Die Ein- und Ausgabeplätze entsprechen den Kanälen im Serviceautomat, die Endmarkierungen den Endzuständen.

Wenn wir von einem oWFN ausgehen, möchten wir möglicherweise den Liberalsten Partner auch als oWFN betrachten. Neben der trivialen Umwandlung des Partners in ein Zustandsnetz, in dem jede Kante durch Transitionen ersetzt wird, welche die entsprechenden Knoten des Automaten als Vor- und Nachplätze erhalten, bietet die Regionentheorie ([Cortadella et al., 1997]) einen probaten Ansatz, aus einem Transitionssystem, wie es ein Automat ist, ein Petrinetz mit möglichst viel Nebenläufigkeit zu generieren.

Durch diese Umformung entsteht ein oWFN, dessen Erreichbarkeitsgraph isomorph zum Liberalsten Partner ist, das heißt, jeder erreichbaren Markierung des oWFN entspricht genau ein Zustand des Liberalsten Partners.

Definition 18 (Endmarkierung im oWFN)
Seien A ein Serviceautomat, O das dazugehörige oWFN und $h : A \to EG(O)$ der Isomorphismus zwischen A und dem Erreichbarkeitsgraphen $EG(O)$ von O. Dann ist $m = h(q)$ genau dann eine Endmarkierung *von O, wenn $q \in Q_A$ ein Endzustand von A ist.* ⌟

Da der Liberalste Partner ein Serviceautomat ist, können wir die Definition auf diesen anwenden. Wir müssen jedoch sicher stellen, dass die Markierungen, die wir als Endmarkierungen identifizieren, gültige Endmarkierungen gemäß unserer Definition von oWFN (Definition 17) ist.

Lemma 3.2 (Endmarkierung im oWFN)
Seien A ein Serviceautomat und O das dazugehörige oWFN. Dann ergibt Definition 18 gültige Endmarkierungen. ⌟

Beweis. Laut Definition 17 müssen Endmarkierungen zwei Bedingungen erfüllen:

- Die Ein- und Ausgabeplätze sind nicht markiert, da im entsprechenden Endzustand keine Nachrichten mehr unterwegs sind.
- Die Markierung aktiviert keine Transition. Anderenfalls gäbe es im Erreichbarkeitsgraphen einen Übergang, der einem internen oder Sendeereignis entspräche. Dieser Übergang müsste aufgrund der Isomorphie auch in A möglich sein, was jedoch der Definition eines Endzustandes widerspricht.

Somit ist jede zu einem Endzustand isomorphe Markierung eine gültige Endmarkierung. □

3.2.3 Endzustände im Public View

Der Begriff *Public View* ist nicht genau definierbar. Wir wollen darunter einen Serviceautomaten verstehen, der zu einem gegebenen Automaten in der Form äquivalent ist, dass er die (bis auf Isomorphie) gleiche Bedienungsanleitung wie der ursprüngliche Automat besitzt. Der Sinn dahinter ist in der Regel das Bedürfnis, interne Strukturen eines Services zu verstecken, also Details, die nicht für die *Öffentlichkeit* bestimmt sind.

Die Frage, die sich uns stellen muss, ist, in welcher Art und Weise wir von einem gegeben Serviceautomaten abstrahieren können, um den Public View zu erhalten – ob es vielleicht einen kanonischen Public View gibt. Da wir aber bereits keine sinnvolle formale Definition angeben können, was wir uns unter einem Public View vorstellen wollen, ist auch dessen direkte Erzeugung aus einem Automaten erstmal nicht vorstellbar.

Jedoch können wir mit Hilfe der Bedienungsanleitungen einen Serviceautomaten generieren, der die im Anfang des Abschnittes geforderte Eigenschaft besitzt und somit als Public View bezeichnet werden kann (da wir von der Bedienungsanleitung ausgehen und nicht vom ursprünglichen Automaten, sind dessen interne Details tatsächlich verborgen).

Definition 19 (Public View)
Seien A ein Serviceautomat und OG dessen Bedienungsanleitung. Dann bezeichnet P den Public View *von A, wenn wir P folgendermaßen konstruieren:*

1. *der duale Automat OG' zu OG wird gebildet (d. h. eingehende und ausgehende Kanäle werden vertauscht, also $I_{OG'} = O_{OG}$ und $O_{OG'} = I_{OG}$), und wir merken uns die final-Annotationen,*
2. *für jeden Zustand q von OG': für jedes $x \in I_{OG'}$, für das kein Übergang in $q \in Q_{OG'}$ existiert, füge einen Übergang x von $q \in Q_{OG'}$ zu einem speziellen Verklemmungszustand in OG' ein,*
3. *für jeden Zustand q von OG': wenn $\phi(q) = !x_1 \vee \ldots \vee !x_k [\vee final] \wedge \ldots$ in OG, dann füge einen neuen Zustand q' (mit der entsprechend gemerkten final-Annotation) in OG' ein mit einem τ-Übergang von $q \in Q_{OG'}$ nach $q' \in Q_{OG'}$ und für alle Übergänge $[q, x, r]$ mit $r \in I_{OG'}$ füge den Übergang $[q', x, r]$ in OG' ein,*
4. *die Endzustände von OG' werden gemäß unserem Kriterium nach Definition 16 gewählt.*

Als P wählen wir dann den nichtannotierten Automaten OG'. ⌟

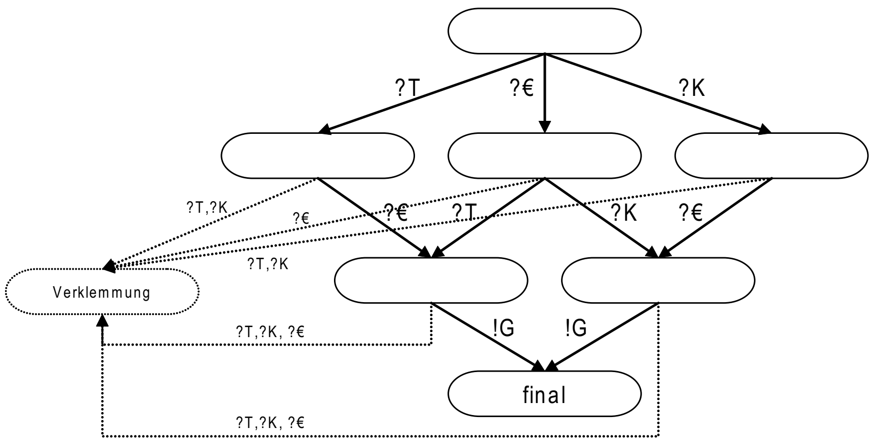

Abbildung 3.1: Aus der Bedienungsanleitung in Abbildung 2.6 ergibt sich der gezeigte Public View des Getränkeautomaten. Die gepunkteten Linien führen zu dem speziellen Verklemmungszustand. Der Zustand, in dem *final* steht, ist der Endzustand des Public View.

Es kann gezeigt werden, dass der auf diese Weise erzeugte Serviceautomat die Bedienungsanleitung besitzt, aus der er erzeugt wurde. (Ein exakter Beweis soll demnächst veröffentlicht werden.) Die Wahl der Endzustände stellt dabei sicher, dass wieder genau die gleichen Knoten die Annotation *final* enthalten, die sie ursprünglich erhalten haben. Genauer gesagt: Für jeden Zustand der Bedienungsanleitung, der als Annotation das Literal *final* enthält, stellt Punkt 3 der Definition sicher, dass im Public View eine Situation erreichbar ist, die eine entsprechende *final*-Annotation bewirkt. Somit ist auch im Falle des Public Views unsere Definition für die Menge der Endzustände sinnvoll gewählt.

4 Fazit

In dieser Arbeit haben wir uns den Formalismus der Serviceautomaten angeschaut und uns intensiv mit dem Aspekt der Endzustände beschäftigt. Wir haben die Begriffe Bedienungsanleitung, Liberalster Partner und Public View eingeführt und für diese eine gemeinsame neue Definition für die Menge der Endzustände eingeführt. Für jeden dieser drei Begriffe konnten wir zeigen, dass die neue Definition sinnvoll ist. Für deren Anwendung konnten wir darlegen, dass sich das veränderte Kriterium problemlos einfügt.

Für das Matching mit einer Bedienungsanleitung sind die Endzustände eh belanglos, für den Liberalsten Partner erhalten wir sogar ein hinreichendes Kriterium für die Wahl der Endzustände und damit einen stärkeren Bezug zum ursprünglich gegebenen Serviceautomaten und eine bessere Intuition. Zuletzt haben wir auch für den Public View plausibel gemacht, dass wir die neue Definition sinnvoll gewählt haben und dass der Umfang der gewählten Menge an Endzustände hinreichend ist.

Die Ergebnisse sollen jetzt noch praktisch umgesetzt werden. In das Werkzeug `Fiona` wird die neue Definition integriert, sodass künftig Bedienungsanleitungen und Public Views mit den entsprechenden Endzuständen automatisiert berechnet werden können.

Bei zukünftigen Anwendungen der Serviceautomaten und der damit verwandten Begriffe wird die gegebene Definition erneut hinterfragt werden.

Literaturverzeichnis

[CKK+97] Jordi Cortadella, Michael Kishinevsky, Alex Kondratyev, Luciano Lavagno, and Alexandre Yakovlev. A region-based theory for state assignment in speed-independent circuits. In *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems, Vol. 16, No. 8, August 1997.* IEEE, 1997.

[LMSW07] Niels Lohmann, Peter Massuthe, Christian Stahl, and Daniela Weinberg. Analyzing Interacting WS-BPEL Processes Using Flexible Model Generation. *Data Knowl. Eng.*, 2007. accepted for special issue of BPM 2006.

[LMW07] Niels Lohmann, Peter Massuthe, and Karsten Wolf. Operating guidelines for finite-state services. In Jetty Kleijn and Alex Yakovlev, editors, *28th International Conference on Applications and Theory of Petri Nets and Other Models of Concurrency, ICATPN 2007, Siedlce, Poland, June 25-29, 2007, Proceedings*, volume 4546 of *Lecture Notes in Computer Science*, pages 321–341. Springer-Verlag, 2007.

[MW06] Peter Massuthe and Karsten Wolf. An Algorithm for Matching Nondeterministic Services with Operating Guidelines. Informatik-Berichte 202, Humboldt-Universität zu Berlin, 2006.