# Modellierung von Ortseinschränkungen für mobile Geschäftsprozesse mit höheren Petri-Netzen

Michael Decker

Institut AIFB, Karlsruher Institut für Technologie (KIT)
m.decker@kit.edu

**Abstract:** Mobile (Geschäfts-)Prozesse sind teilgeordnete Folgen von Aktivitäten zur Erreichung eines bestimmten Ziels (z.B. Auslieferung einer Bestellung, Reparatur einer technischen Anlage), bei denen einzelne Aktivitäten mit einem mobilen Computer (z.B. Smartphone, Netbook) ausgeführt werden. Im Beitrag wird ein Ansatz beschrieben, Einschränkungen für die zulässigen Ausführungsorte einzelner Prozessaktivitäten in mit höheren Petri-Netzen beschriebenen Prozessmodellen zu definieren. Diese sog. *Ortseinschränkungen* berücksichtigen auch prozess-spezifische Merkmale, etwa dass innerhalb einer Prozessinstanz verschiedene Aktivitäten am selben Ort ausgeführt werden müssen.

## 1 Einleitung

Bei der Realisierung *prozesszentrierter Informationssysteme (IS)* werden die Abarbeitungsfolgen von Aktivitäten analysiert, die zur Erledigung bestimmter Aufgaben erforderlich sind. Die Menge der Aktivitäten zur Erledigung einer Aufgabe ist teilgeordnet und wird üblicherweise von mehrere Akteuren ausgeführt. Im Idealfall kann ein solches IS die Abarbeitung von Prozessen proaktiv unterstützen, indem es etwa die einzelnen Aktivitäten automatisch geeigneten Akteuren zuordnet und die rechtzeitige Erledigung überwacht [Obe05]. Ein Beispiel für einen solchen (Geschäfts-)Prozess wäre etwa die Bearbeitung einer Kundenbestellung, bei der zuerst verschiedene Prüfungen durchzuführen sind (z.B. bzgl. Bonität des Kunden, Verfügbarkeit der gewünschten Ware), und im Falle positiver Ergebnisse die weiteren Aktivitäten für den eigentlichen Versand (Kommissionierung, Druck der Versandpapiere, Prüfung der Sendung, Verpacken & Verladen) durchgeführt werden.

Viele in der Praxis zu findende Prozesse beinhalten Aktivitäten, die typischerweise außerhalb der Reichweite von stationären Computersystemen durchgeführt werden, z.B. Tätigkeiten vor Ort bei Kunden wie Beratung, Wartung, Auftragsakquise oder Datenerfassung. Dank der Fortschritte auf dem Gebiet der *mobilen Technologien* können auch Prozesse mit solchen mobilen Aktivitäten durchgehend von IS unterstützt werden. Zu den mobilen Technologien zählen neben den verschiedenen Arten von mobilen Computern (z.B. Smartphones und Netbooks) auch drahtlose Datenkommunikation (z.B. WLAN oder UMTS) und Ortungssysteme (z.B. GPS oder Zellortung in Mobilfunknetzen). Ein *mobiler Prozess* liegt dann vor, wenn er mindestens eine Aktivität umfasst, die unter Verwendung eines

mobilen Computers ausgeführt wird.

Um den spezifischen Merkmalen solcher Prozesse gerecht zu werden, sind besondere Modellierungsansätze notwendig. Im vorliegenden Beitrag wird deshalb das Konzept der „Ortseinschränkungen“ (OE) eingeführt, mit dem Aussagen über den Ausführungsort einzelner Aktivitäten in einem Prozess gemacht werden können. Damit wird die Mobilität der Akteure und somit das besondere Merkmal dieser Art von Prozessen adressiert. Diese OE werden auf mit Petri-Netzen dargestellten Prozessmodellen angewendet.

Die Idee ortsabhängiger Zugriffskontrolle für mobile Informationssysteme wird von einigen Arbeiten aufgegriffen (z.B. [DBP07]), die sich aber auf Modelle ohne Prozesswissen beschränken. In den meisten Arbeiten wird dabei eine Erweiterung der „Role-based Access Control“ (RBAC) um ortsabhängige Elemente vorgenommen, z.B. in der Form, dass komplette Rollen oder einzelne Berechtigungen „ausgeschaltet“ werden, wenn sich der jeweiligen Nutzer außerhalb eines festgelegten Gebiets befindet. Prozess-spezifische Beschränkungen – wie etwa, dass zwei Aktivitäten innerhalb einer Prozessinstanz am selben Ort ausgeführt werden müssen – können mit solchen Modellen nicht abgebildet werden.

Die Definition und Durchsetzung von Ortseinschränkungen für bestimmte Aktivitäten kann durch Sicherheitsaspekte motiviert sein: die Gefahr des Missbrauchs eines abhanden gekommenen mobilen Computers durch den Dieb oder unehrlichen Finder kann reduziert werden, wenn bspw. bestimmte Funktionen außerhalb des Firmengeländes oder anderer Orte gesperrt wird. Das Ausspähen von Daten „über die Schulter“ wird verhindert, wenn OE verbieten, sensitive Daten an öffentlichen Orten abzurufen. Kann der mobile Computer zur Fernsteuerung stationärer Anlagen (z.B. Gebäude- oder Hörsaaltechnik) verwendet werden, so soll dies nur möglich sein, wenn der Akteur sich in unmittelbarer Nähe zu der Anlage befindet. Weiter kann es Compliance-Anforderungen geben, die mit OE durchgesetzt werden können, wenn etwa bestimmte Software-Funktionen aus Lizenzgründen nur innerhalb bestimmter Bereiche verwendet werden dürfen („Campus-Lizenz“) oder Mitarbeiter die Anwesenheit an einem bestimmten Ort nachweisen müssen. Aber auch Usability-Aspekte können OE motivieren, wenn an bestimmten Orten irrelevante Daten und Optionen automatisch verborgen werden, um so die Interaktion mit der ohnehin eingeschränkten Nutzerschnittschnelle mobiler Computer zu vereinfachen.

Der verbleibende Teil des vorliegenden Artikels ist wie folgt gegliedert: Im nächsten Abschnitt werden zunächst die für das Verständnis erforderlichen Grundlagen aus dem Bereich der Petri-Netze vorgestellt. Aufbauend hierauf können dann in Abschnitt 3 verschiedene Arten von Ortseinschränkungen eingeführt und anhand von Petri-Netzen formal definiert werden; dies schließt auch die Beschreibung eines geeigneten Ortsmodells ein. Zur Demonstration der Anwendung der vorgeschlagenen Modellierungstechnik werden in Abschnitt 4 zwei zusammenhängende Prozesse vorgestellt. In Abschnitt 5 werden Arbeiten anderer Autoren vorgestellt, die mit dem hier vorgestellten Ansatz verwandt sind, bevor der Artikel mit einem Fazit in Abschnitt 6 endet.

## 2 Prozessmodellierung mit Petri-Netzen

Petri-Netze sind ein grafischer Formalismus zur Beschreibung von Abläufen und Systemen [Rei10]. In der Angewandten Informatik werden Petri-Netze u.a. zur Modellierung von Geschäftsprozessen eingesetzt. Gegenüber anderen üblicherweise für die Prozessmodellierung eingesetzten Notationen wie UML Aktivitätendiagrammen, Ereignisgesteuerten Prozessketten oder der „Business Process Modeling Notation“ (BPMN) zeichnen sich Petri-Netze durch ihre mathematische Fundierung aus. Wir verzichten in diesem Papier allerdings soweit wie möglich auf formale Definitionen und beschränken uns auf aussagekräftige Beispiele.

In Abbildung 1 ist ein Petri-Netz, oder genauer ein sog. „Stellen-Transitionen-Netz“, abgebildet: Ein solches Netz in ein gerichteter Graph mit zwei Arten von Knoten: „Stellen“ werden als Kreise und „Transitionen“ als Vierecke dargestellt. Pfeile verbinden Stellen mit Transitionen bzw. Transitionen mit Stellen. Stellen können Marken — dargestellt durch schwarze Punkte — enthalten, sind also als „Behälter anonymer Objekte“ interpretierbar. Die Transitionen stellen Aktivitäten oder Vorgänge dar; eine Transition kann „ausgeführt“ werden, was als „Schalten“ bezeichnet wird, wenn alle Eingangsstellen mindestens eine Marke enthalten. Das Schalten selbst ist ein atomarer Vorgang, bei dem aus jeder Eingangsstelle eine Marke entfernt wird und in jede Ausgangsstelle eine Marke eingefügt wird. In Abbildung 1 könnte etwa die Transition $t4$ schalten, da in jeder ihrer beiden Eingangsstellen $s3$ und $s4$ sich jeweils eine Marke befindet. Aber auch eine der beiden Transitionen $t2$ und $t3$ könnte schalten, da sich in der gemeinsamen Eingangsstelle $s2$ eine Marke befindet. Welcher Schaltvorgang zuerst ausgeführt wird ist nicht vorgegeben. Da bei der gegebenen Markierung nur eine der beiden Transitionen $t2$ und $t3$ schalten kann, konkurrieren diese beiden Transitionen um die Marke in $s2$.

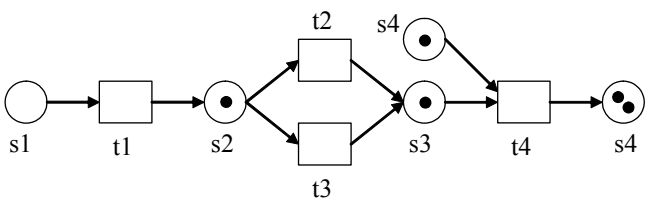


Abbildung 1: Beispiel für ein Stellen-Transitionennetz mit Marken

Bei sog. „höheren Petri-Netzen“ sind die Marken keine anonymen sondern unterscheidbare Objekte. Ein Beispiel für höhere Netze sind „Prädikate/Transitionen-Netze“ (PrT-Netze) [GL81], bei denen die Stellen Relationsschemata (Prädikate) repräsentieren. Die einzelnen „Marken“ können also als Datensätze in einer Tabelle aufgefasst werden. Einer Transition kann ein optionaler prädikatenlogischer Ausdruck zugewiesen werden, der zusätzlich erfüllt sein muss, damit die Transition „aktiviert“ wird und schalten kann.

In Abbildung 2 ist ein Beispiel für eine Transition in einem PrT-Netz zu sehen: es gibt die drei Stellen „Bestellungen“, „Teilelager“ und „Pack-Aufträge“, deren zugehörige Re-

lation jeweils in Form einer Tabelle wiedergegeben sind. Eine Bestellung besteht aus einer Bestellnummer und umfasst immer genau ein Teil, das durch die Angabe der Teile-Nummer spezifiziert wird. In der Tabelle „Teilelager“ ist für jeden belegten Lagerplatz angegeben, welche Teile-Nummer das dort gelagerte Teil hat; in jedem Lagerplatz kann sich höchstens ein Teil befinden. Die Transition „Packauftrag erzeugen“ entnimmt aus jeder der beiden Eingangstabellen je einen Datensatz (Tupel); hierzu sind die Eingangspfeile mit Variablentupeln beschriftet, deren Wertigkeit der Anzahl der Spalten der zugehörigen Stelle entsprechen muss. Diese Transition trägt auch einen einfachen prädikatenlogischen Ausdruck ($b = c$) der fordert, dass die Teile-Nummern der aus den beiden Stellen entnommenen Datensätzen übereinstimmen. Wenn diese Transition schaltet, dann werden die beiden Datensätze aus den Eingangsstellen entfernt und es wird ein neuer Datensatz in der Ausgangsstelle „Pack-Aufträge“ eingefügt: jeder Datensatz in dieser Tabelle gibt Auskunft darüber, auf welchen Lagerplatz für die Erfüllung eines Auftrags zugegriffen werden muss.

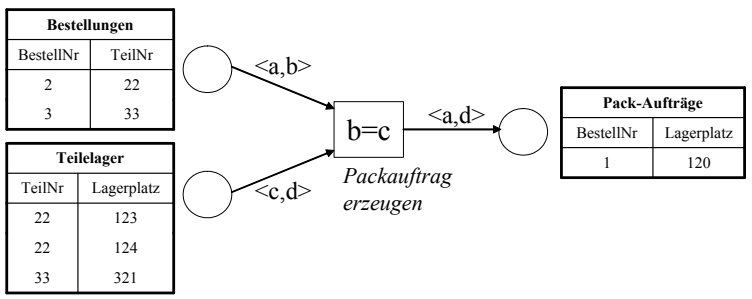


Abbildung 2: Beispiel für ein PrT-Netz

# 3 Ortseinschränkungen

## 3.1 Ortsmodell

Bevor die verschiedenen Formen von Ortseinschränkungen (OE) eingeführt werden können, muss erst noch das zugrunde liegende Ortsmodell beschrieben werden. Ein Ortsmodell ist ein spezielles Datenmodell zur Beschreibung räumlicher Daten, die im vorliegenden Fall auf zwei Dimensionen beschränkt sind. Es können also keine übereinander liegenden Orte (z.B. mehrere Stockwerke in einem Gebäude) abgebildet werden; die Erweiterung des folgenden Modells auf drei Dimensionen ist aber einfach möglich.

Der aktuelle Aufenthaltsort eines (mobilen) Akteurs wird als Punkt angenommen, der etwa über GPS- oder Zell-ID-Ortung festgestellt wird. Flächen werden durch Polygone mit mindestens drei Eckpunkten beschrieben, wobei es nicht zulässig ist, dass sich zwei Linien eines Polygons überkreuzen – es muss sich also um „einfache Polygone“ handeln.

Jedes Polygon ist genau einem Ortstyp zugeordnet. Ortstypen stellen eine Schema-Ebene zur Klassifikation der durch die Polygone beschriebenen Orte dar. Beispiele für Ortstypen könnten etwa „Land“ oder „Stadt“ sein, aber auch anwenderspezifische Typen wie „Abteilung“ oder „Verkaufsbezirk“ sind denkbar. Jedes Polygon kann als (Orts-)Instanz des zugehörigen Ortstyps aufgefasst werden, wobei es nie zwei Instanzen des gleichen Ortstyps geben kann, die den selben Punkt beinhalten. Dieser Ansatz findet sich auch in der „Geographic Markup Language“ (GML) in Form von „Features“, die immer genau einem „Feature-Type“ zugeordnet sind [Bur06]. Auch in gängigen Geo-Informationssystemen (GIS) findet sich dieser Ansatz in Form verschiedener „Ebenen“ (Layer), die nur Objekte eines Typs enthalten (z.B. Layer1: „bebaute Flächen“; Layer2: „Gewässer“). Für die Formulierung der Transitionsinschriften werden noch einige Prädikate, Funktionen und Konstanten für die Arbeit mit räumlichen Daten benötigt:

- Die Funktion $getPos(pid)$ liefert die aktuelle Position des aktuellen Akteurs als Punkt im zweidimensionalen Raum zurück. Rein technisch entspricht dies der Abfrage der Ortung (etwa GPS-Empfänger oder Fremdortungssystem). Als Argument muss dieser Funktion die *Prozess-ID (PID)* der aktuellen Prozessinstanz übergeben werden, z.B. $getPos(123)$.
- Ortsinstanzen sind Konstanten, deren Bezeichner vollständig in Grossbuchstaben geschrieben sind, z.B. $BERLIN$, $DEUTSCHLAND$ oder $LAGERHALLE3$. Eine weitere Ortsinstanz namens $WORLD$ beinhaltet die komplette Referenzfläche, d.h. jede andere Ortsinstanz ist komplett in $WORLD$ enthalten.
- Das Prädikat $liegtIn(x, L)$ liefert genau dann den Wert „True“, wenn die Position $x$ sich innerhalb der Ortsinanz $L$ befindet, z.B. $liegtIn(getPos(123), BERLIN)$.
- Ortstypen sind ebenfalls Konstanten, aber die Bezeichner werden mit einem „T_“ für „Typ“ eingeleitet, z.B. $T\_VERKAUFSBEZIRK$ oder $T\_STADT$.
- Die Funktion $getOInstanz(x, t)$ liefert eine Ortsinstanz vom Typ $t$ zurück, die die aktuelle Aufenthaltsposition $x$ des Akteurs beinhaltet. Der folgende Ausdruck liefert bspw. die Ortsinstanz zurück, in der sich der aktuelle Akteur der Prozessinstanz 123 gerade aufhält: $getOInstanz(getPos(123), T\_STADT)$.

### 3.2 Direkte Ortseinschränkungen

Ortseinschränkungen sind Aussagen darüber, wo eine bestimmte Prozessaktivität *ausgeführt werden muss* (positive Einschränkung) oder *nicht ausgeführt werden darf* (negative Einschränkung). Negative Einschränkungen sind vor allem dann von Vorteil, wenn es einfacher ist, explizit alle Ortsinstanzen aufzuzählen, an denen eine Aktivität verboten ist, als alle, wo sie zulässig ist; dies kann etwa dann gegeben sein, wenn eine bestimmte Aktivität nur in einigen wenigen Ländern nicht ausgeführt werden darf, z.B. weil dort Industriespionage zu befürchten ist oder die nationale Gesetzgebung die Durchführung dieser Aktivität nicht gestattet (z.B. weil hierzu der Einsatz von kryptographischen Methoden notwendig wäre).

In Abbildung 3 ist die positive Ausprägung einer direkten OE zu finden: die Aktivität „Auftrag erfassen“ darf nur innerhalb der Ortsinstanz „Berlin“ ausgeführt werden. Dies ist im linken Teil der Zeichnung vereinfacht dargestellt: ein Parallelogramm mit dem Bezeichner der Ortsinstanz zeigt mit einem gestrichelten Pfeil auf die eingeschränkte Transition, wobei auf dem Pfeil in einem Kreis der „Modus“ (positive Einschränkung) durch ein „=“ angegeben ist. OE werden durch gestrichelte Pfeile Transitionen zugewiesen, wobei diese Pfeile *KEINE* Marken transportieren können. Die Transition in diesem Beispiel hat jeweils eine Ein- und eine Ausgangsstelle, deren Schemata nur die Prozess-ID einer Prozessinstanz beschreiben.

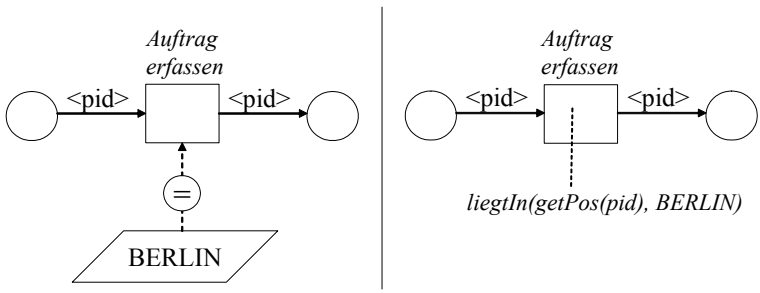


Abbildung 3: Direkte Ortseinschränkung

Im rechten Teil der Abbildung 3 ist diese direkte OE als Prädikat für die Transition formuliert: der Ausdruck $liegtIn(getPos(pid), BERLIN)$ fragt zunächst mit der Funktion $getPos()$ den aktuellen Ort des aktuellen Akteurs ab und überprüft dann mit dem Prädikat $liegtIn()$, ob sich dieser Ort innerhalb der Ortsinstanz $BERLIN$ befindet; nur wenn dies der Fall ist ergibt der Ausdruck den Wert „TRUE“ und die Transition kann schalten.

Die Idee von Ortseinschränkungen für einzelne Aktivitäten in Prozessmodellen im Fall von UML Aktivitätsdiagrammen findet sich auch in [HK09]: die abgerundeten Rechtecke, die eine Aktivität in UML repräsentieren, werden mit jeweils einem kleinem Quadrat gekennzeichnet, das den Bezeichner des zulässigen Orts enthält; negative OE sind nicht vorgesehen. Da jede Aktivität diese Dekoration erhält, müssen Aktivitäten ohne Einschränkung mit einem „$*$“ gefüllt werden. Weiter beschreibt dieser Ansatz kein Ortsmodell und sieht auch keine weiteren Ortseinschränkungen für das Prozessmodell vor.

### 3.3 Indirekte Ortseinschränkungen

Bei den im letzten Abschnitt beschriebenen direkten OE wurden zur *Entwurfszeit* des Prozessmodells die konkreten Ortsinstanzen festgelegt, an denen einzelne Aktivitäten ausgeführt werden müssen oder nicht dürfen. Es sind aber auch Fälle denkbar, in denen sich erst zur *Laufzeit* des Prozesses bestimmen lässt, an welchen Orten eine Aktivität ausgeführt werden darf. In diesem Fall kann im Prozessgraph dann nur dargestellt werden, wie sich der konkrete Ort bestimmen lässt. Eine solche Ortseinschränkung wird daher als „indirekt“

bezeichnet. Direkte OE legen daher Einschränkungen auf der Schema-Ebene des Prozesses fest, während indirekte OE dies auf Instanzenebene tun.

### 3.3.1 Ortsregeln: selber Ort

Der von uns derzeit primär verfolgte Ansätze für indirekte OE sind sog. *„Ortsregeln" (OR)*: hierbei wird in Abhängigkeit des tatsächlichen Ausführungsortes einer vorangegangenen (Quell-)Aktivität eine OE für eine Zielaktivität festgelegt. Es gibt positive OR, die fordern, dass die Zielaktivität am selben Ort ausgeführt wird wie die Quellaktivität (Ortsbindung); eine negative OR hingegeben verbietet, dass die Zielaktivität am selben Ort ausgeführt wird wie die Quellaktivität (Ortstrennung).

Diese beiden Formen von OR entstehen durch die Übertragung der beiden bekannten Sicherheitsprinzipien *„Separations of Duties" (SoD)* und *„Binding of Duties" (BoD)*, die ein wesentliches Merkmal spezieller Zugriffskontrollmodelle für Workflow-Systeme sind (z.B. [WBK03]). Bei SoD wird gefordert, dass zwei Aktivitäten innerhalb einer Prozessinstanz von unterschiedlichen Akteuren ausgeführt werden; z.B. darf der Akteur, der einen Genehmigungsworkflow gestartet hat, nie die Aktivität „Entscheidung treffen" ausführen. BoD hingegen fordert, dass zwei Aktivitäten innerhalb derselben Prozessinstanz auch vom selben Akteur ausgeführt werden, etwa um Kunden einen einheitlichen Ansprechpartner zu bieten („One face to the customer"); wenn Akteur X als die Aktivität „Kundenanfrage entgegennehmen" ausgeführt hat, dann muss er für diese Prozessinstanz auch die Aktivität „Antwort an Kunde mitteilen" ausführen. OR ergeben sich, wenn bei SoD und BoD anstelle der Trennung bzw. Bindung des Akteurs für eine Aktivität die Ausführungsorte für verschiedene Prozessinstanzen getrennt bzw. gebunden werden; in Anlehnung an die Begriffe SoD und BoD kann man deshalb auch von „Separation of Locations" und „Binding of Locations" sprechen.

Ortsbindungen sind oft motiviert durch verschiedene Feldaktivitäten mobiler Arbeiter, die nur am selben Ort sinnvoll sind; etwa kann die Aktivität „Reparatur" nur dort durchgeführt werden, wo auch die ursprüngliche Inspektion stattgefunden hat. Solche OR können auch helfen, Verwechselungen von gleichartigen Anlagen zu vermeiden. Eine Ortstrennung kann bspw. dadurch motiviert sein, dass gewährleistet werden soll, dass Aktivitäten von räumlich getrennten Akteuren ausgeführt werden, damit diese sich nicht persönlich kennen und deshalb bei Fehlern gegenseitig decken; oder um bei Stichproben die erforderliche Streuung zu erzwingen.

Es muss jetzt noch präzisiert werden, was der „selbe Ort" ist: handelt es sich hierbei um dasselbe Gebäude oder dasselbe Land? Zur Spezifikation dieser Granularität des Ortes wird ein Ortstyp genannt, z.B. $T\_PARZELLE$, wenn Quell- und Zielaktivität in derselben Parzelle stattfinden sollen.

Im linken Teil von Abbildung 4 ist eine positive OR abgebildet: die durch die Transition $t1$ repräsentierte Aktivität soll das Grundstück (Ortsinstanz) bestimmen, an dem auch die später folgende Aktivität $t2$ durchgeführt wird. Wieder ist die OE durch einen gestrichelten Pfeil dargestellt, der von der Quellaktivität auf die Zielaktivität zeigt. Auf diesem Pfeil ist wie bei direkten OE der Modus in einem Kreis eingetragen; an diesen Kreis an-

gebracht ist auch ein Rechteck, welches den Bezeichner des Ortstyps für die Spezifikation der Granularität trägt.

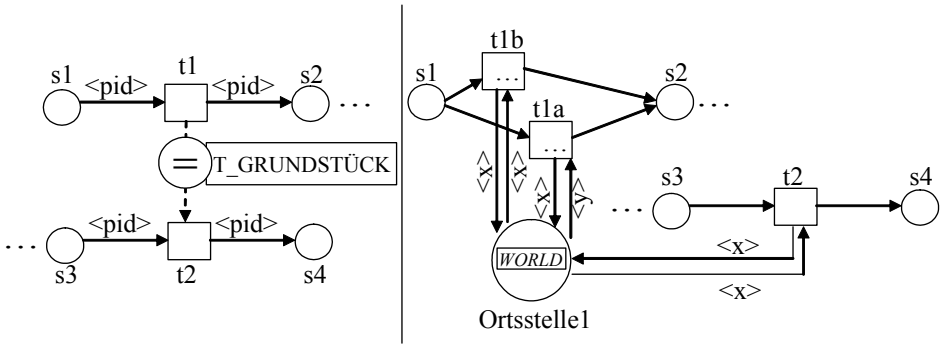


Abbildung 4: Indirekte Ortseinschränkung

Um diese OR mit einem PrT-Netz zu formalisieren, wird die Transition $t1$ durch zwei Transitionen $t1a$ und $t1b$ ersetzt. Dies ist rechts in Abbildung 4 dargestellt, wobei der Übersichtlichkeit wegen auf die Beschriftung der Pfeile mit „$< pid >$“ verzichtet wurde. Zur Speicherung des Ortes wird eine Ortsstelle eingeführt, die initial die Ortsinstanz $WORLD$ enthält, so dass der Ausführungsort von $t2$ zunächst überhaupt nicht eingeschränkt ist. Die Zielaktivität $t2$ darf nur schalten, wenn sich der Akteur innerhalb der in der Ortsstelle gespeicherten Ortsinstanz befindet, was durch die folgende Transitionsinschrift für $t2$ festgelegt wird: $liegtIn(getPos(pid), x)$. Da beide Pfeile, die $t2$ mit der Ortsstelle verbinden, mit „$< x >$“ beschriftet sind, ändert ein Schalten von $t2$ nicht den Inhalt der Ortsstelle, die Ortseinschränkung wird also nicht „verbraucht“, sondern nur gelesen.

Die Transition $t1a$ soll nur bei der allerersten Ausführung der Quellaktivität schalten, dann aber in der Ortsstelle $WORLD$ durch die Ortsinstanz ersetzen, die dem Grundstück entspricht, in dem sich der Akteur gerade befindet. Wir formlieren deshalb folgende Transitionsinschrift für $t1a$:

$$y = WORLD \wedge\ x = getOInstanz(getPos(pid), T\_GRUNDST\ddot{U}CK)$$

Bei allen folgenden Ausführungen der Quellaktivität soll $t1b$ schalten, welche den Inhalt der Ortsstelle nicht ändert. Die Inschrift für diese Transition lautet deshalb: $x \neq WORLD$.

### 3.3.2 Ortsregeln mit Zuordnungslisten

Ortsregeln können auch dann zum Einsatz kommen, wenn der Quell- und Zielort *nicht* identisch sind. Unser Ansatz sieht deshalb auch OR vor, die sog. Zuordnungslisten verwenden, mit denen jedem Quellort ein Zielort zugeordnet wird. Die durch diese Liste beschriebene Zuordnung muss nicht injektiv sein, es kann also vorkommen, dass zwei oder

mehr Quellorte auf den gleichen Zielort abgebildet werden. Mit dieser Art von OR lassen sich also Regeln der Art „wenn Aktivität $A1$ an Ort $O1$ ausgeführt wird, dann muss Aktivität $A2$ an $O2$ ausgeführt werden“ (mit $O1 \neq O2$) formulieren. Ein Anwendungsfall hierfür ist die Formulierung von regionalen Zuständigkeiten, bei denen die aus einem bestimmten Gebiet eingegangenen Aufträge in einer bestimmte Filiale bearbeitet werden müssen.

Abbildung 5 befasst sich mit dieser Art von OE. Im linken Teil ist die informelle Notation dargestellt: der gestrichelte Pfeil dieser OR ist mit einem Kreis gekennzeichnet, der einen Implikationspfeil ($\Rightarrow$) trägt. Anstelle eines Ortstyps ist der Bezeichner der Zuordnungsliste angegeben, der in einem Rechteck mit einer „Wellenlinie“ als untere Seite enthalten ist. Der rechte Teil von Abbildung 5 zeigt die Darstellung dieser OR als PrT-Netz: für die Zuordnungsliste $L\_LISTE1$ gibt es eine eigene Stelle, neben der auch einige Einträge aufgeführt sind, wobei die linke Spalte die Quell- und die rechte die Zielorte enthält. Aus Platzgründen kann die Zielaktivität in Form von Transition $t2$ nicht dargestellt werden. Abgesehen hiervon unterscheidet sich die Darstellung nur durch die zwei Pfeile zwischen $t1a$ und der Stelle für die Liste. Diese Transition kann nur schalten, wenn die Ortsstelle noch den initialen Wert „WORLD“ beinhaltet. Beim Schalten wird aus der Zuordnungsliste das Tupel „$< a, b >$“ entfernt (und gleich wieder eingefügt), bei dem $a$ den Aufenthaltsort des Akteurs enthält. Die Inschrift von $t1a$ lautet deshalb wie folgt:

$$y = WORLD \wedge\ liegtIn(getPos(pid), a)$$

Wenn unter Erfüllung dieses Ausdrucks die Transition schaltet, dann enthält die Variable $b$ den entsprechenden Zielort, der in die Ortsstelle eingefügt wird.

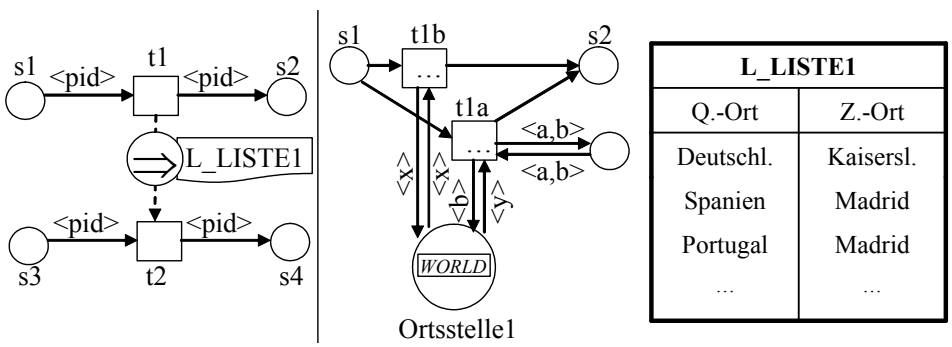


Abbildung 5: Ortsregel mit Zuordnungsliste

### 3.3.3 Abgekürzte Schreibweisen

Es sollen noch zwei abkürzende Schreibweise für OE beschrieben werden, die in Abbildung 6 dargestellt sind: Im linken Teil hat die mit „Entnahme“ beschriebene Transition eine direkte OE, mit den beiden Ortsinstanzen $HALLE1$ und $HALLE2$. Da es sich um

eine negative OE handelt, darf die Entnahme-Aktivität nicht durchgeführt werden, solange der Akteur sich in einer der beiden Hallen befindet. Die Inschrift für die Transition lautet also wie folgt:

$$\neg(\ liegtIn(getPos(pid), HALLE1)\ \vee\ liegtIn(getPos(pid), HALLE2)\ )$$

Im rechten Teil von Abbildung 6 ist eine Ortsregel mit zwei Zielaktivitäten zu sehen. Die OR erzeugt positive Ortseinschränkungen des Ortstyps $T\_ABTL$, die also einzelnen Abteilungen in einem Unternehmen entsprechen. Sowohl $t4$ als auch $t5$ müssen folglich in der Abteilung ausgeführt werden, in der zuvor auch $t1$ ausgeführt wurde. Formal wird dies dadurch realisiert, dass beide Transitionen in ihrer Inschrift den Inhalt der von $t1$ verwalteten Ortsstelle auswerten.

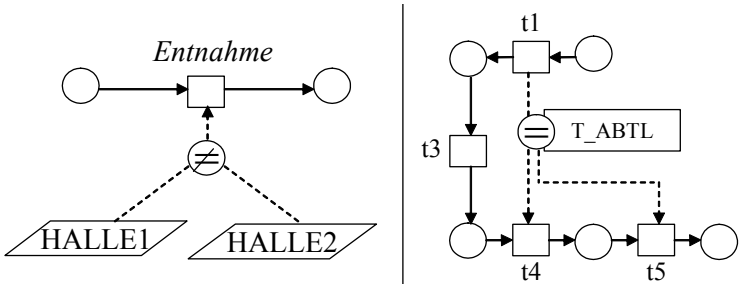


Abbildung 6: Direkte OE mit mehreren Orten (links) und Ortsregel mit zwei Zielaktivitäten (rechts)

## 4 Szenarien

Zur Demonstration der Anwendung der vorgeschlagenen Methode zur Modellierung von mobilen Prozessen mit Ortseinschränkungen wird zunächst die Auslieferung von Ersatzteilen für Fabrikanlagen durch einen spezialisierten Anbieter betrachtet. Dieser Anbieter ist außer in Deutschland auch noch in anderen europäischen Ländern tätig, bietet aber den betrachteten Dienst nur in Deutschland an. Zur Organisation der logistischen Abwicklung hat die Firma Deutschland in Regionen unterteilt. In jeder Region gibt es genau ein Lager sowie mehrere Filialen für die Betreuung der Kunden. Neben der Lieferung von Ersatzteilen bietet die Firma weitere Dienstleistungen wie die Durchführung von Wartungsarbeiten für Fabrikanlagen an, so dass ein über ganz Europa ausgebreitetes Netz an Filialen erforderlich ist, von dem aus Servicetechniker entsendet werden können.

Der betrachtete Prozess ist in Abbildung 7 dargestellt, wobei für die OE aus Platzgründen die vereinfachte Form gewählt wurde. Um die Übersichtlichkeit zu wahren, wurde von den Stellen nur die für „Start“ und „Ende“ beschriftet. Die erste Aktivität ist die Entgegennahme des Anrufs einer Fabrik bei der zuständigen Filiale. In jeder Filiale kann über ein ERP-System noch während des Telefonats festgestellt werden, ob das gewünschte Ersatzteil im zuständigen Regional-Lager aktuell verfügbar ist; ist dies nicht der Fall, so

wird ein hier nicht betrachteter Prozess gestartet, um das Ersatzteil direkt vom Produzenten zur Fabrik zu transportieren. Diese Aktivität wird mit direkten OE auf Filialen in Deutschland beschränkt. Nach der Erfassung der Bestellung in der Filiale wird diese über das Workflow-System des Unternehmens automatisch an das zuständige Lager weitergeleitet. Da die einzelnen Lager immer für eine bestimmte Region zuständig sind, kann mit einer Zuordnungsliste abgeleitet werden, welches Lager zuständig ist. In diesem Lager werden die Ersatzteile kommissioniert und in ein Lieferfahrzeug verladen. Im Idealfall ist die nächste Aktivität eine erfolgreiche Lieferung. Es ist aber auch möglich, dass eine Zustellung nicht möglich ist, weil die Warenannahme der Fabrik nicht besetzt ist oder wegen eines Fehlers die Annahme verweigert wird. In diesem Fall kann eine erneute Lieferung zu einem späterem Zeitpunkt versucht werden; diese muss am selben Ort erfolgen, was durch eine Ortsregel mit „Erfolglose Zustellung“ als Quelle und den beiden Ziel-Transitionen „Erfolgreicher Lieferversuch“ und „Erfolgloser Lieferversuch“ ausgedrückt wird. Nach einem fehlgeschlagenen Lieferversuch kann aber auch entschieden werden, die Lieferung zurück in das Lager zu bringen; eine entsprechende Ortsbindung legt fest, dass dies das Ursprungslager der Ware sein muss. Nachdem die Lieferung im Ursprungslager wieder eingebucht wurde, muss auch der Kunde telefonisch verständigt werden; über eine weitere Ortsregel wird gewährleistet, dass dieser Anruf von der Filiale aus getätigt wird, die auch die ursprüngliche Bestellung erhalten hat. Auch wenn die Lieferung dem Kunden nicht zugestellt werden kann hat dieser die Kosten zu tragen; vor Ende der Prozessinstanz wird also in jedem Fall eine Rechnung erstellt. Dies ist wieder Aufgabe der zuständigen Filiale, was ebenfalls durch eine Ortsregeln definiert ist.

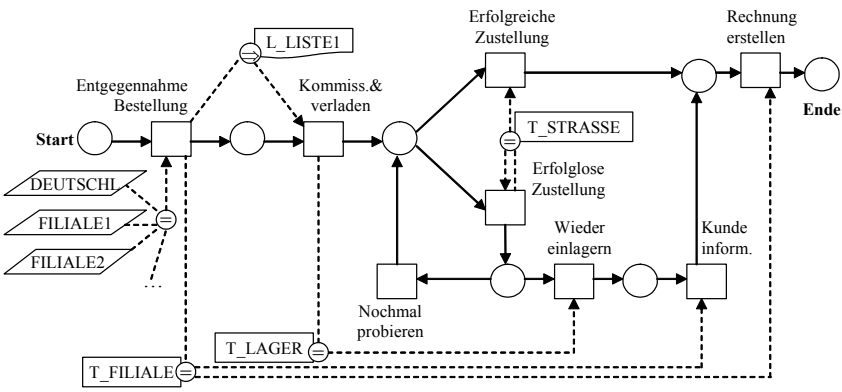


Abbildung 7: Prozessgraph für Szenario „Ersatzteil-Lieferung“

Da dieses Szenario keine negativen OE beinhaltet, soll noch ein weiterer Prozess betrachtet werden (Abbildung 8): der Prozess findet sich in einem multinationalem Unternehmen, welches auf die Durchführung chemischer Analysen spezialisiert ist, worunter auch Bodenproben fallen. Hierzu betreibt es mehrere Labore in verschiedenen europäischen Ländern. Eine Instanz dieses Prozesses beginnt wieder mit der Entgegennahme eines Auftrages. Allerdings ist dies nicht in $LAND1$ möglich, da hier aufgrund der starken Konkur-

renzsituation diese Analysedienstleistung nicht zu einem kostendeckenden Preis angeboten werden kann. Es werden dann von Angestellten der Firma von zwei unterschiedlichen Feldern innerhalb einer Gemeinde zwei Bodenproben genommen. Durch zwei Ortsregeln mit unterschiedlichem Modus ist gewährleistet, dass dies innerhalb derselben Gemeinde geschieht, aber auf verschiedenen Feldern. Damit die Qualität der chemischen Analyse gewährleistet ist, werden die Proben 1 und 2 jeweils in A- und B-Probe aufgeteilt, die dann in getrennten Laboren analysiert werden müssen. Die beiden Aktivitäten „A-Probe auswerten“ und „B-Probe auswerten“ sind deshalb mit zwei symmetrischen Ortsregeln versehen, die verhindern, dass die Auswertungen im selben Labor der Firma durchgeführt werden. Es sind hierfür zwei entgegengesetzte OR notwendig, da nicht vorhergesagt werden kann, welche der beiden Analysen zuerst durchgeführt wird. Wenn beide Ergebnisse vorliegen, werden diese konsolidiert. Es kann dann zur letzten Aktivität der Prozessinstanz, nämlich der Erstellung von Bericht und Rechung, übergegangen werden. Diese Aktivität kann in jeder Niederlassung der Firma erstellt werden, außer solchen, die sich in $LAND2$ oder $LAND3$ befinden, da hier die notwendige Software aus Kostengründen nicht lizenziert wurde.

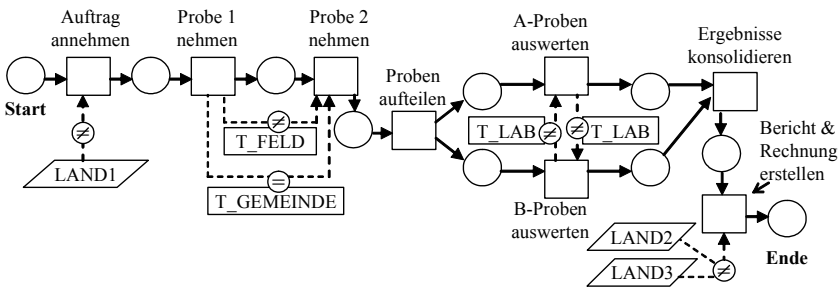


Abbildung 8: Prozessgraph für Szenario „chemische Analyse von Bodenproben“

## 5 Verwandte Arbeiten

Im Workflow-Management gibt es den Ansatz des „Constraint-based Workflow Modelings“ (CBWM, z.B. [SOS05]). Die Einschränkungen (Constraints) bei CBWM machen eine Aussage über die Ablaufreihenfolge der Aktivitäten und nicht über zulässige Ausführungsorte. CBWM wurde entwickelt, um flexiblere Prozesse (z.B. Ausnahmebehandlungen, Sonderwünsche von Kunden) zu ermöglichen. Eine Aktivitätsfolge ist zulässig, solange sie nicht gegen eine der formulierten Einschränkungen verstößt.

In [KMR03] werden Petri-Netze verwendet, bei denen die einzelnen Stellen diskrete Orte (z.B. einen Raum) darstellen. Das Schalten einer Transition entspricht also einer Ortsänderung. Eine solche Stelle kann selbst wieder ein Petri-Netz enthalten, weshalb dieser Ansatz „Nets within Nets“ genannt wird. Die Netze innerhalb einer Stelle repräsentieren

den Zustand eines Haushaltsroboters.

Baumeister et al. beschäftigen sich auch mit der Darstellung von Mobilität in Prozessmodellen [BKKW02]; sie verwenden allerdings Aktivitätendiagramme aus der UML. Ihre Erweiterung dieser Diagramme zielt aber nicht darauf ab, Ortseinschränkungen für die Ausführungen von Aktivitäten zu formulieren, sondern zu modellieren, wie durch Aktivitäten der Ort von physischen Objekten verändert wird. Unter Verwendung dieser Notation müsste die Aktivität „Ersatzteil zustellen“ mit dem Stereotyp ≪move≫ ausgezeichnet werden, da durch ihre Ausführung der Ort eines Objekts verändert wird.

In [KG04] wird die „Mobile Landscaping“-Methode vorgestellt, welche u.a. die Darstellung der Verteilungsstruktur von mobilen Prozessen mit elementaren Petri-Netzen beinhaltet. Durch das Unterlegen von Teilen von Petri-Netzen mit Rechtecken wird dargestellt, welche Aktivitäten von welcher Organisation oder von welchen Mitarbeitern ausgeführt werden, Hierauf aufbauend können in einem weiteren Schritt mobile Prozessteile identifiziert werden. Dieser Ansatz beinhaltet aber nicht die Definition von Ortseinschränkungen.

Wenn die Ortung eines mobilen Akteurs Grundlage für eine Zugriffskontrollentscheidung ist, stellt sich die Frage, inwieweit das verwendete Ortungssystem gegen externe und interne Angreifer resistent ist. Verschiedene Ansätze, um solche Location-Spoofing-Angriffe zu verhindern oder zu erkennen, werden in [Dec09] vorgestellt.

Es gibt einige Arbeiten, die spezielle *Workflow-Managementsysteme (WfMS)* mit Unterstützung für mobile Akteure vorschlagen (siehe [DKOS10] für einen Überblick). Eine mobil-spezifische Funktion eines solchen m-WfMS könnte etwa sein, anstehende Aktivitäten einer Prozessinstanz dem Akteur zuzuordnen, der sich gerade in geringster Entfernung zum Ausführungsort befindet; oder die Workflow-Client-Anwendung für den mobilen Computer implementiert Mechanismen, um auch bei zeitweiser Nichtverfügbarkeit der drahtlosen Internetanbindung autark Workflow-Items abarbeiten zu können. Die Idee von OE findet sich in Arbeiten über m-WfMS aber nicht.

## 6 Zusammenfassung und Ausblick

Im vorliegenden Artikel wurde eine auf Petri-Netzen basierende Methode zur Definition von *Ortseinschränkungen (OE)* in Geschäftsprozessmodellen eingeführt. Da der aktuelle Aufenthaltsort eines Nutzers bei der Abarbeitung einer Prozessaktivität der mobilspezifischste Kontextparameter ist, eignet sich diese Methode insbesondere für mobile Geschäftsprozesse.

Auch wenn der vorgeschlagene Ansatz die Definition vielfältiger OE ermöglicht, sind trotzdem noch Geschäftsregeln denkbar, die mit den hier vorgestellten Konstrukten nicht abgebildet werden können, z.B. „wenn Aktivität $A1$ in $L1$ und Aktivität $A2$ in $L2$ ausgeführt wurde, dann muss $A3$ in $L3$ ausgeführt werden“. Als weitere Möglichkeit sind deshalb auch sog. „externe OE“ als Spezialfall indirekter OE vorgesehen, bei denen auf ein externes System verwiesen wird, das zur Laufzeit dann die konkrete OE errechnet. Diese Berechnung kann im externen System von einer im Prinzip beliebig komplexen Programmlogik durchgeführt werden.

# Literatur


[BKKW02] Hubert Baumeister, Nora Koch, Piotr Kosiuczenko und Martin Wirsing. Extending Activity Diagrams to Model Mobile Systems. In *Proceedings of NetObjectDays (NOD)*, Seiten 278–293, Erfurt, 2002. Springer-Verlag.

[Bur06] David S. Burggraf. Geography Markup Language. *Data Science*, 5:178–204, 2006.

[DBP07] Maria Luisa Damiani, Elisa Bertino und Paolo Perlasca. Data Security in Location-Aware Applications: An Approach Based on RBAC. *International Journal of Information and Computer Security*, 1(1/2):5–38, 2007.

[Dec09] Michael Decker. Ein Überblick über Ansätze zur Vermeidung der Manipulation von Ortungsverfahren. In *Proceedings zur 4. Konferenz „Mobile und ubiquitäre Informationssysteme“(GI-MMS 2009)*, Seiten 53–66, Münster, 2009.

[DKOS10] Michael Decker, Björn Keuter, Andreas Oberweis und Peter Stürzel. Workflow-Management mit Mobile Computing: Ein Überblick. In *Proceedings des Fachgespräch „Ortsbezogene Anwendungen und Dienste“(2009)*, Seiten 145–154, Bonn, 2010.

[GL81] Hartmann J. Genrich und Kurt Lautenbach. System Modelling with High-Level Petri Nets. *Theoretical Computer Science*, 13(1):109–136, 1981.

[HK09] Rattikorn Hewett und Phongphun Kijsanayothin. Location Contexts in Role-based Security Policy Enforcement. In *Proceedings of the 2009 International Conference on Security and Management (SAM'09)*, Seiten 404–410, Las Vegas, Nevada, USA, 2009.

[KG04] André Köhler und Volker Gruhn. Mobile Process Landscaping am Beispiel von Vertriebsprozessen in der Assekuranz. In *Proceedings der Konferenz MCTA 2004*, Seiten 12–24, Augsburg, 2004.

[KMR03] Michael Köhler, Daniel Moldt und Heiko Rölke. Modeling Mobility and Mobile Agents Using Nets within Nets. In *Proceedings of ICATPN 2003*, LNCS, Seiten 121–139, Eindhoven, Netherlands, 2003. Springer-Verlag.

[Obe05] Andreas Oberweis. *Process-Aware Information Systems. Bridging People and Software Through Process Technology*, Kapitel Person-to-Application Processes: Workflow Management, Seiten 21–36. John Wiley & Sons, New York, USA, et al., 2005.

[Rei10] Wolfgang Reisig. *Petrinetze: Modellierungstechnik, Analysemethoden, Fallstudien*. Vieweg+Teubner, Wiesbaden, 2010.

[SOS05] Shazia W. Sadiq, Maria E. Orlowska und Wasim Sadiq. Specification and validation of process constraints for flexible workflows. *Information Systems*, 30(5):349–378, 2005.

[WBK03] Jacques Wainer, Paulo Barthelmess und Akhil Kumar. W-RABC – A Workflow Security Model Incorporating Controlled Overriding of Constraints. *International Journal of Cooperative Information Systems*, 12(4):455–485, 2003.