Karlsruher Institut für Technologie

Petri Net Markup Language

Seminararbeit

von
Johannes Reiß

an der Fakultät für
Wirtschaftswissenschaften

in dem Studiengang
Informationswirtschaft

eingereicht am 3. Juli 2013 beim
Institut für Angewandte Informatik
und Formale Beschreibungsverfahren
des Karlsruher Instituts für Technologie

Referent: Prof. Andreas Oberweis
Betreuer: David Karlin

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Die Modellierung von Abläufen ist eine wesentliche Voraussetzung bei der Entwicklung von Software oder der Anlagenkonzeption im Maschinenbau. Aber auch das Management von Geschäftsprozessen und Workflows in Unternehmen profitiert durch gezielte und genaue Dokumentation der Abläufe. Dabei ist es in einer vielschichtigen IT-Landschaft wichtig, dass einheitliche Standards und Austauschformate zwischen verschiedenen Werkzeugen existieren und unterstützt werden. Diese Aufgabenstellung wurde in einem Workshop über „XML/SGML basierte Austauschformate für Petri-Netze“ [BBK+00] aufgegriffen und ein erster Vorschlag zur *Petri Net Markup Language (PNML)* erarbeitet. [WK03, HKK+09]

In der Arbeit wird nach einer kurzen Einführung in Petri-Netze auf das Konzept der PNML eingegangen und deren Elemente anhand von Beispielen vorgestellt. Bevor zum Schluß der Arbeit ein Beispiel-Geschäftsprozess den Zusammenhang zwischen Petri-Netzen und der PNML verdeutlicht, wird zuvor noch auf die Erweiterungsmöglichkeiten der PNML eingangen.

2 Einführung in Petri-Netze

Die ursprüngliche Einführung von Petri-Netzen geht auf die Dissertationschrift von Carl Adam Petri über „Kommunikation mit Automaten“ [Pet62] zurück, welche er am 27.07.1961 an der Universität Bonn einreichte. Die nachfolgenden formalen Definitionen eines Petri-Netzes und dessen Elementen haben sich mit kleineren Unterschieden in der Literatur etabliert, hierzu vergleiche man beispielsweise [KKB05], [Rei86].

Zunächst wird in Abschnitt 2.1 die formale Definition von Petri-Netzen vorgestellt und welche Bedingung für einzelne Elemente gelten müssen. Weiter wird in Abschnitt 2.2 die gebräuchliche Darstellung von Petri-Netzen als Graphenrepräsentation vorgestellt.

2.1 Formale Definition

Ein einfaches Petri-Netz ist aus zwei Elementen aufgebaut, den *Transitionen* und den *Stellen*, die häufig auch *Places* genannt werden. Im allgemeinen werden diese beiden Elemente durch eine Relation miteinander verbunden. Man spricht in diesem Zusammenhang auch von der *Flussrelation*.

Bezeichnen wir nun eine Menge an *Stellen* als S mit den einzelnen *Stellen* $s_1, s_2, \ldots, s_n$ und eine Menge an *Transitionen* als T mit den einzelnen *Transitionen* $t_1, t_2, \ldots, t_m$. Damit einzelne *Stellen* mit einer *Transition* oder eine *Transition* mit einer *Stelle* verbunden werden kann definieren wir die *Flussrelation* als Relation $F \subseteq (S \times T) \cup (T \times S)$. Weiter muss gelten, dass eine *Stelle* nicht auch *Transition* sein kann, also gilt: $S \cap T = \emptyset$. Ein Petri-Netz besteht außerdem aus mindestens einer *Stelle* oder einer *Transition*, es gilt: $S \cup T \neq \emptyset$. Das vollständige Petri-Netz setzt sich zusammen zu einem Tupel $N = (S, T, F)$. Um eine wesentliche Eigenschaft von Petri-Netzen, die Simulation von Abläufen nützen zu können, definieren wir eine Funktion $m : S \to \mathbb{N}_0$. Sie weist jeder *Stelle* aus S eine Anzahl von Marken zu. Diese Funktion wird

Markierung genannt.

Betrachten wir nun beispielhaft ein Petri-Netz. Dieses Netz besitzt vier *Stellen*, die wir $s1$, $s2$, $s3$, $s4$ nennen. Weiter besitzt das Petri-Netz noch drei *Transitionen* $t1, t2, t3$. Somit sind dann unsere Mengen $S = \{s1, s2, s3, s4\}$ und $T = \{t1, t2, t3\}$. Nach obiger Definition ist dies bereits ein Petri-Netz. Möchte man einen Ablauf modellieren, dann fehlen dazu noch Verbindungen zwischen den *Stellen* und *Transitionen*. Hierzu bilden wir mit Hilfe der *Flussrelation* die folgenden Paare aus jeweils einer *Stelle* und einer *Transition* $(s1, t1)$, $(s2, t2)$, $(s3, t3)$, $(t1, s2)$, $(t1, s3)$, $(t2, s4)$, $(t3, s4)$. Das Paar $(s1, t1)$ bedeutet zum Beispiel, dass die *Stelle* $s1$ mit der *Transition* $t1$ verbunden ist und zwar von $s1$ nach $t1$. Es soll außerdem noch die erste *Stelle* $s1$ eine Marke beinhalten, also $m(s1) = 1$. Man schreibt dies auch als sogenannte *Markierungsfolge* $(1, 0, 0, 0)$. Dies bedeutet, dass die *Stelle* $s1$ eine Marke beinhaltet und die *Stellen* $s2, s3, s4$ jeweils keine Marken beinhalten.

Für das gewünschte Modell eines Ablaufs benötigen wir noch eine Definition wann eine *Transition* schalten soll. Hierzu müssen zuerst die Begriffe des Vor- und Nachbereichs einer *Transition* t definiert werden. Sei t eine *Transition* und s eine *Stelle*, dann ist der *Vorbereich* von t die Menge der *Stellen* für die $\{s \mid (s, t) \in F\}$ gilt. Also alle *Stellen*, die eine gerichtete Verbindung *zur Transition* t haben. Man schreibt hierfür auch $\cdot t$. Der *Nachbereich* wird analog als Menge $\{s \mid (t, s) \in F\}$ definiert und als $t\cdot$ notiert. Eine *Transition* t kann also schalten, wenn für alle *Stellen* $s \in \cdot t$ folgendes gilt: $m(s) \geq 1$. Nachdem *Transition* t geschalten hat, ändert sich die Markierung des Petri-Netzes wie folgt: Die Anzahl der Marken in den *Stellen* des Vorbereichs werden um eins verringert, also $\forall s \in \cdot t : m'(s) = m(s) - 1$. Die Anzahl der Marken in den *Stellen* des Nachbereichs werden um jeweils eins erhöht, also $\forall p \in t\cdot : m'(p) = m(p) + 1$. Die Anzahl in den *Stellen*, die weder im *Vorbereich* noch im *Nachbereich* von t liegen wird nicht verändert.

Das Petri-Netz hat nun *Stellen*, *Transitionen*, eine *Flussrelation*, eine Markierung und eine Regel, wann *Transitionen* schalten. Da die textuelle Beschreibung, aber nicht sehr anschaulich ist hat sich zur besseren Darstellung von Petri-Netzen die Repräsentation als Graph etabliert, worauf im nächsten Abschnitt eingegangen wird.

2.2 Repräsentation als Graph

Nachdem im vorangegangen Abschnitt die Elemente eines Petri-Netzes kurz vorgestellt worden sind, sollen in diesem Abschnitt die einzelnen Elemente und das eingeführte Beispiel veranschaulicht werden.

Betrachten wir die *Stellen*, diese werden in der Abbildung als Kreis dargestellt. Ist eine *Stelle* markiert, so wird diese *Stelle* mit einem Punkt innerhalb des Kreises gekennzeichnet. Sollte die Markierung mehrere Marken vorsehen, erhält der Kreis so viele Punkte, wie Marken vorgesehen sind. Zur besseren Zuordnung kann die *Stelle* mit einem Bezeichner versehen werden. *Transitionen* werden in der Regel durch ein Rechteck dargestellt. Manchmal sind sie aber auch der Einfachheit halber als senkrechte Striche ausgeführt. Bezeichner oder Inschriften sind ebenfalls möglich. Die Abbildung 1 zeigt eine *Stelle* und *Transition* eines Petri-Netzes. Die Elemente der *Flussrelation* werden durch Kanten zwischen den jeweiligen *Stellen* und *Transitionen*

dargestellt.

Abbildung 1: Stelle s mit zwei Marken und Transition t

In dem Beispiel aus Abschnitt 2.1 war das Petri-Netz wie folgt definiert: $S = \{s1, s2, s3, s4\}$, $T = \{t1, t2, t3\}$, $F = \{(s1, t1), (s2, t2), (s3, t3), (t1, s2), (t1, s3), (t2, s4), (t3, s4)\}$ und der Anfangsmarkierung $m = (1, 0, 0, 0)$. Daraus ergibt sich dann der Graph aus Abbildung 2. Wie zu erkennen ist, gibt es keine direkte Kante zwischen zwei *Transitionen* oder zwei *Stellen*. Man nennt diese Eigenschaft auch *Bipartitheit*, sie ergibt sich aus der Tatsache, dass $S \cap T = \emptyset$ gilt.

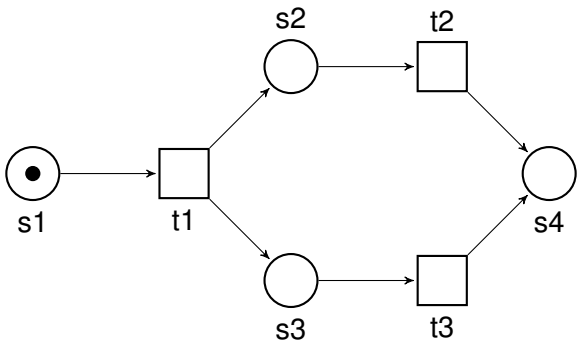

Abbildung 2: Das Beispielnetz aus Abschnitt 2.1

Neben der hier vorgestellten Definition mit reinen *Stellen* und *Transitionen* existieren weitere Arten von Petri-Netzen für die unterschiedlichsten Zwecke. Als Beispiele sind gefärbte Petri-Netze (Coloured Petri Nets) in [JK09] oder stochastische Netze in [AM95] genannt. Gefärbte Petri-Netze und andere Erweiterungen werden auch als höhere Petri-Netze (High-Level Petri Nets) bezeichnet. Höhere Petri-Netze werden in *ISO/IEC 15909* und dem zweiten Teil *15909-2* standardisiert. Auf diesen baut die Petri Net Markup Language, die im nächsten Abschnitt betrachtet werden soll, auf.

3 Petri Net Markup Language

Nachdem im Abschnitt 2 kurz die wesentlichen Elemente eines Petri-Netzes vorgestellt und erklärt wurden, soll dieser Abschnitt einen Einblick in die Konzepte der Petri Net Markup Language geben, sowie deren Elemente und das Mapping auf XML vorstellen. Abschließend wird auf Erweiterungsmöglichkeiten der Petri Net Markup Language eingegangen.

3.1 Konzept und Metamodell

Die zugrunde liegende Idee von Austauschformaten im Allgemeinen ist, dass man Daten aus einem Programm mit anderen Programmen, meist systemübergreifend, austauschen möchte. Nun sind aber Datenverarbeitungssysteme zuweilen sehr heterogen und haben unterschiedliche Anforderungen zu erfüllen. Dies macht die Entwicklung eines geeigneten Austauschformats nicht einfach und stellt gewisse Anforderungen an das Austauschformat.

Zur Konkretisierung der Anforderungen betrachtet man zuerst die Probleme, die bei Anwendung eines Austauschformats auftreten können. Auf Hinblick von Petri-Netzen muss hier vor allem die Erweiterbarkeit genannt werden, also ob das Austauschformat auch mit neueren Typen von Petri-Netzen, welche bei der Formatentwicklung noch nicht bekannt waren, kompatibel ist. Ein weiteres Problem ist die Offenheit für möglichst viele Plattformen und Systeme. Es müssen also Technologien eingesetzt werden, die nicht an bestimmte Betriebssysteme oder Rechnerarchitekturen gebunden sind. Außerdem kann es sein, dass durch den Datenaustausch Informationen verloren gehen könnten, dies ist zu vermeiden oder muss zumindest reduziert werden. Die genannten Probleme versuchte man bei der Entwicklung der Petri Net Markup Language zu vermeiden und hat hierzu die nachfolgenden Prinzipien aufgestellt.

Damit möglichst alle Typen von Petri-Netzen und wichtige Eigenschaften abgebildet werden können, soll das Austauschformat flexibel sein. Die PNML realisiert dies, indem Petri-Netze als Graphen mit Inschriften, sogenannten *Labels*, verstanden werden. [BCH+03] Die Inschriften ermöglichen es, dass aus noch unbekannten Typen von Netzen Informationen gewonnen werden können. [WK03]

Neben der Flexibilität soll die PNML auch Kompatibilität bieten. Wie bereits beschrieben, ist es möglich auch mit Beschreibungen von Netzen zu arbeiten, deren Typ dem Programm nicht bekannt ist. Hierzu werden sogenannte *Conventions* eingesetzt. *Conventions* sind eine erweiterbare Menge an vordefinierten Inschriften und deren Bedeutung. [BCH+03]

Mit dem Prinzip der Kompatibiltät hängt das Prinzip der Eindeutigkeit zusammen. Es muss sichergestellt werden, dass aus der Repräsentation eines Petri-Netzes in PNML eindeutig der zugrunde liegende Typ bzw. das zugrunde liegende Petri-Netz bestimmt werden kann. [BCH+03]

Außerdem sollte das Format menschenlesbar und mit einem Texteditor bearbeitbar sein. [WK03] Man hat sich dafür entschlossen auf eine Repräsentation durch die Extensible Markup Language (XML) zu setzen. Diese syntaktische Repräsentation ist aber zuersteinmal unabhängig vom eigentlichen Konzept. [WK03]

Das eigentliche Konzept ist vielmehr eine Struktur mit vielen einzelnen Elementen, die miteinander im Zusammenhang stehen. Zum Beispiel gibt es sogenannte *Petri Net Files*, diese müssen bestimmte Voraussetzungen erfüllen, die in der *Petri Net Type Definition* festgelegt sind, welche sich wiederum auf ein *Conventions Dokument* beziehen. Diese grobe Struktur ist aber eigentlich wesentlich feiner aufgebaut. So sind in einem *Petri Net File* mehrere Petri Netze möglich, die wiederum aus unterschiedlichen Objekten und Inschriften aufgebaut sein können. Objekte sind zum Beispiel Kanten und Knoten (*Stellen* und *Transitionen*). In einem *Petri Net File* sind aber auch Angaben von Werkzeugen zu finden. Werkzeuge für die Modellierung von Petri-Netzen mit Unterstützung der PNML haben die Möglichkeit eigene Angaben in die PNML-

Datei zu schreiben. Hierzu bedienen sie sich speziell gekennzeichneter XML-Elemente.

Neben den genannten Elementen, die zusammen das Modell für die „basic PNML“ [BCH+03] bilden, gibt es noch *Referenzknoten* (reference nodes), die bei Petri-Netzen über mehrere Seiten (pages) Verwendung finden. Diese weiteren Elemente eines Petri Net Files gehören zur „structured PNML“ [BCH+03]. Auf die einzelnen Elemente im Detail soll im nächsten Abschnitt eingegangen werden.

3.2 Elemente

Im vorherigen Abschnitt wurden bereits wesentliche Elemente der Petri Net Markup Language aufgezählt und kurz deren Zusammenhang aufgezeigt. Nun soll genauer auf die einzelnen Elemente eingegangen werden.

Betrachtet man zuerst das *Petri Net File*. Dieses kann auch mit einem Container verglichen werden, welchen man mit Inhalt beladen kann. Im Falle von Petri Net Files ist die Ladung mindestens ein Petri-Netz. Da auch mehrere Petri-Netze innerhalb eines *Petri Net File* möglich sind, ist ein eindeutiger Bezeichner (ID) für jedes Petri-Netz vorgesehen. Außerdem ist die Angabe des Typs für jedes Petri-Netz notwendig. Dies geschieht durch Angabe eines Uniform Resource Identifier (URI) der *Typdefinition*. Auf der Website des Projekts [1] finden sich zum Beispiel Definitionen von Stellen/Transitions-Netzen oder High-Level Petri-Netzen.

Ein Petri-Netz besteht wiederum aus verschiedenen Objekten (objects), wobei es sich in der Regel um die Elemente der Graphrepräsentation handelt. [WK03] Wie man im Abschnitt 2.2 über die graphische Repräsentation gesehen hat, gibt es im wesentlichen vier verschiedene Arten von Objekten: Knoten in Form von *Stellen* und *Transitionen*, Kanten und Inschriften. Alle Objekte haben einen eindeutigen Bezeichner zur Identifikation und graphische Informationen. Bei den *Stellen* und *Transitionen* wird als graphische Information die Position des Knotens angegeben, bei Kanten ist es eine Liste von Punkten, durch welche die Kante verläuft.

Die Inschriften werden nochmals in zwei Unterobjekte aufgeteilt: *Annotationen* und *Attribute*. Bei Annotationen wird die relative Position zum zugehörigen Objekt als grafische Information angegeben. Attribute haben dagegen keine grafischen Informationen, da sie nicht als Text zum zugehörigen Objekt erscheinen. *Annotationen* sind also Inschriften, die Anmerkungen, als Text, an einem bestimmten Objekt darstellen. Beispiele für *Annotationen* sind Namen von Knoten oder Bezeichner von Kanten. Eventuelle Marken von *Stellen* können ebenfalls durch *Annotationen* abgebildet werden. [WK03] Attribute hingegen sind meist Eigenschaften von Objekten, die nicht textuell wiedergegeben werden, wie Farbe oder Form eines Objekts. [WK03] Inschriften sind aber nicht zwingend an Objekte, wie Knoten oder Kanten gebunden, sie können auch bei Petri-Netzen oder bei Seiten verwendet werden, sofern Letztere definiert sind.

Wie oben erwähnt wurde ist eines der Prinzipien der PNML die Kompatibilität. Darum ist in der PNML die Möglichkeit vorhanden werkzeugspezifische Informationen zu definieren. Diese Information wird zusätzlich zu den anderen Informationen gespeichert und im Format klar kenntlich gemacht. Hierzu wird das Werkzeug und dessen Version angegeben. So ist gewährleistet,

[1] `http://www.pnml.org/version-2009/version-2009.php`

dass andere Werkzeuge diese Informationen erkennen können und nicht zwingend verarbeiten müssen.

Diese Elemente der „basic PNML“ können durch *Referenzknoten* und Seiten zu „structured PNML“ erweitert werden. Für besonders große Petri-Netze kann es sinnvoll sein, dass diese in kleinere Einheiten unterteilt werden. In etwa so, wie wenn man ein Petri-Netz von Hand zeichnet und dieses nicht vollständig auf das vorhandene Blatt passt. Seiten gehören wie Knoten, Kanten und Inschriften auch zu der Gruppe der Objekte. Weitere Unterseiten von Seiten sind möglich.[HKK+09] Wenn man sich das Beispiel mit dem Blatt vor Augen führt, sieht man, dass es Schwierigkeiten bereitet eine Kante, ohne Unterbrechungen, zwischen zwei Blättern zu zeichnen. Auch in der PNML ist dies nicht möglich. Die Verbindung der Knoten erfolgt anhand eines Referenzknotens, der den jeweiligen Knoten auf der anderen Seite referenziert. Dabei muss beachtet werden, dass die Typen der *Referenzknoten* gleich sind, also das man eine *Stelle* auch mit einer Referenzstelle abbildet und eine Referenztransition auch eine *Transition* referenziert. [HKK+09]

Name		XML-Tag	Attribut
Petri Netz Dokument		<**pnml**>	
Petri Netz		<**net**>	ID Typ
Objekt			
	Seite	<**page**>	ID
	Stelle	<**place**>	ID
	Transition	<**transition**>	ID
	Kante	<**arc**>	ID Quelle Senke
	Referenzstelle	<**referencePlace**>	ID Referenz
	Referenztransition	<**referenceTransition**>	ID Referenz
Inschrift			
	Annotation Attribut		
Werkzeuginformation		<**toolspecific**>	Werkzeug Version

Tabelle 1: Elemente in der PNML und dazugehörige XML-Elemente [BCH+03, Table 1]

Die beschriebenen Elemente bilden die Grundlage der PNML. Es fehlt allerdings noch eine geeignete Darstellung, die der Anforderung der Menschenlesbarkeit und das einfache Verändern im Texteditor ermöglicht. Bereits bei der Entwicklung des Formats hat man sich für XML entschieden. Die Tabelle 1 gibt einen Überblick der vorgestellten Elemente, deren Bezeichnung in XML und deren Attribute. Die genaue Umsetzung in XML wird im folgenden Abschnitt genauer erklärt.

3.3 Mapping auf XML

Bevor man die einzelnen Elemente durch XML-Ausdrücke abbildet, soll hier kurz der Aufbau von XML erläutert werden. Für die Umsetzung der PNML werden im wesentlichen XML-Elemente benutzt. Ein XML-Element hat zwei sogenannte *Tags*, die den Ausdruck umgeben. Nimmt man als Element zum Beispiel ein Buch, dann sieht der Ausdruck in XML wie folgt aus: `<buch>...</buch>`. Die XML-Elemente können auch Attribute haben, um das Element näher zu spezifizieren. Das Buch kann zum Beispiel einen Typ haben, den man folgendermaßen angeben kann: `<buch typ=sachbuch>...</buch>`. Es ist natürlich auch möglich den Typ als Subelement anzugeben: `<buch><typ>...</typ>...</buch>`.

Im nächsten Schritt wird betrachtet, wie die im vorangegangen Abschnitt vorgestellten Elemente der PNML in XML repräsentiert werden. Es ist sinnvoll vom allgemeinsten Element anzufangen, da man dadurch besser die Struktur des XML-Dokuments verstehen kann .

Das *Petri Net File* wird als XML-Element `<pnml>...</pnml>` abgebildet, weil es sich um das erste Element im XML-Dokument handelt, hat es noch ein Attribut `xmls=""`, das den Namensraum (Namespace) für das XML-Dokument festlegt.

```
<pnml xmlns="http://example.com/pnml-namespace/example">...</pnml>
```

Als nächstes Element der PNML kommt das Petri-Netz an sich. Es wird einfach nur mit `<net>` bezeichnet und besitzt als Attribute eine ID und einen Typ.

```
<net id="sampleNet" type="URI of an Net Type Definition">...</net>
```

Sofern mit Seiten gearbeitet wird, kann auf das Element `<page>` mit einer entsprechenden ID zurückgegriffen werden.

```
<page id="samplePage">...<page>
```

Die Elemente *Stellen* `<place>` und *Transitionen* `<transition>` benötigen jeweils eine ID als Attribut. Zusätzlich enthalten sie als Subelement `<graphics>`-Element, welches wiederum selbst nochmals Kindelemente enthält. Außerdem wird als *Annotation* der Name der *Stelle* mit `<name>` und `<value>` angegeben. Für die *Annotation* ist ebenfalls der Einsatz des `<graphics>`-Elements möglich. Im Beispiel sieht man den Offset, als relative Positionsangabe der *Annotation* zur *Stelle*.

```
<place id="samplePlace">
  <graphics>
    <position x="x-coordinate" y="y-coordinate"/>
  </graphics>
  <name>
    <value>Name der Stelle</value>
    <graphics>
      <offset x="x-coordinate" y="y-coordinate"/>
```

```
      </graphics>
    </name>
  </place>
```

Zusätzlich ist es noch möglich die Markierung der *Stelle* anzugeben. Hierzu benutzt man das XML-Element <**initialMarking**> und gibt als Wert die Anzahl der Marken der *Stelle* an.

```
<place id="samplePlace">
  ...
  <initialMarking>
    <value>Marke</value>
    <graphics>
      <offset x="x-coordinate" y="y-coordinate"/>
    </graphics>
  </initialMarking>
  ...
</place>
```

Transitionen werden, wie bereits erwähnt, durch <**transition**> abgebildet. Wie auch bei den Stellen wird hier ebenfalls eine ID als XML-Attribut angegeben und grafische Informationen in Form des <**graphics**>-Elements ausgezeichnet.

```
<transition id="sampleTransition">
  <graphics>
    <position x="x-coordinate" y="y-coordinate"/>
  </graphics>
</transition>
```

Die Verbindungen der einzelnen Knoten erfolgt durch eine Kante als <**arc**>-Element. Bei einem <**arc**>-Element wird die Quelle (source) und die Senke (target) der Kante jeweils als XML-Attribut angegeben. Hier gibt man für das <**arc**>-Element ebenfalls eine ID an. Da bei Auswahl der Quelle und Senke nicht zwischen *Stellen* und *Transitionen* unterschieden wird, ist es möglich, dass auch Kanten zwischen *Stellen* und *Stellen* bzw. *Transitionen* und *Transitionen* abgebildet werden. Bei der formalen Definition von Petri-Netzen in Abschnitt 2 wird dies ausgeschlossen. Die PNML benötigt diese Möglichkeit, um Kanten zu den *Referenzknoten* abbilden zu können und weil es letzendlich auch von der entsprechenden *Petri Net Type Definition* abhängt, ob solche Netze erlaubt sind. [BCH+03]

```
<arc id="sampleArc" source="samplePlace" target="sampleTrans">
  ...
</arc>
```

Als grafische Information werden beim <**arc**>-Element Punkte angegeben, durch welche die Kante verlaufen soll.

```
<arc id="sampleArc" source="samplePlace" target="sampleTrans">
```

```
    <graphics>
      <position x="x-coordinate of point 1"
            y="y-coordinate of point 1"/>
      <position x="x-coordinate of point 2"
            y="y-coordinate of point 2"/>
    </graphics>
  </arc>
```

Neben diesen grundlegenden Elementen bietet die PNML aber auch für die werkzeugspezifischen Informationen und allgemeine Beschriftungen XML-ELemente an. Spezifische Informationen eines Werkzeugs können durch das <**toolspecific**>-Element in das Dokument eingebaut werden. Dabei muss der Name des Werkzeugs und die Version angegeben werden. Werkzeugspezifische Informationen hängen aber immer vom entsprechenden Werkzeug ab und werden von diesem definiert, weshalb keine allgemeine Dokumentation der möglichen Subelemente existiert.

Das nachfolgende Beispiel ist dem Listing 2 aus [BCH+03] entnommen. Das Element <**hidden** /> kann, je nachdem wie es das Werkzeug „PN4all“ interpretiert, zum Beispiel definieren, dass das entsprechende Element der PNML nicht angezeigt wird.

```
<toolspecific tool="PN4all" version="0.1">
  <hidden/>
</toolspecific>
```

Allgemeine Beschriftungen von Elementen können mit dem <**inscription**>-Element realisiert werden. Der eigentliche Text der Beschriftung wird im Element <**value**> angegeben. Die Position und sonstige grafische Eigenschaften werden durch das bekannte <**graphics**>-Element definiert.

```
<inscription>
  <value>Beschriftungstext</value>
  <graphics>
    <offset x="x-coordinate" y="y-coordinate">
  </graphics>
</inscription>
```

Mögliche Referenzen auf *Stellen* oder *Transitionen* sind analog zu den eigentlichen *Stellen* bzw. *Transitionen* definiert. Zum Einsatz von Referenzstellen benutzt man das XML-Element <**referencePlace**>. Zusätzlich zum ID-Attribut wird eine Referenz zur gewünschten *Stelle* mit dem Attribut **ref** angegeben.

```
<referencePlace id="sampleRefPlace" ref="samplePlace">
  ...
</referencePlace>
```

Der Inhalt des XML-Elements bestimmt sich analog zu dem Inhalt eines normalen Stellenelements. Referenztransitionen werden durch das <**referenceTransition**>-Element festgelegt und enthalten neben der ID die entsprechende Referenz.

```
<referenceTransition id="sampleRefTransition"
                     ref="sampleTransition">
  ...
</referenceTransition>
```

Weiter oben hat man die XML-Elemente <**initialMarking**> für *Stellen* und <**inscription**> für Beschriftungen gesehen. Diese sind ein Beispiel für die Funktionsweise einer *Typdefinition*. Im eigentlichen Modell kommen die beiden XML-Elemente nicht vor. *Typdefinitionen* erlauben jedoch die Definition zusätzlicher XML-Elemente. Unter einer *Typdefinition* versteht man eine XML-Grammatik, die die Struktur und Elemente eines Petri-Netz Dokuments definiert. Der *Typdefinition* vorgelagert stehen die *Conventions*, welche ebenfalls eine XML-Grammatik ist. Für das *Conventions Dokument* und die *Typdefinition* wurde ursprünglich *TREX*[2] als Schemasprache verwendet [WK03], diese wurde später durch *RELAX NG*[3] ersetzt. [BCH+03].

Sollen nun Elemente definiert werden, die in der bisherigen *Typdefinition* nicht vorkommen, so erstellt man eine neue *Typdefinition*, die, je nach Bedarf, bereits vorhandene *Typdefinitionen* importiert. Anschließend können die neuen Elemente definiert werden. Im Rahmen des nächsten Abschnitts über die Erweiterungsmöglichkeiten der PNML wird auf diese Möglichkeit näher eingegangen.

3.4 Erweiterungsmöglichkeiten

Die Elemente der PNML wurden in den vorherigen Abschnitten genauer vorgestellt. In diesem Abschnitt sollen Möglichkeiten zur Erweiterung der PNML betrachtet werden. Zuerst geht man dazu zur Grundlage der PNML zurück, dem Petri-Netz. Als das Petri-Netz am Anfang eingeführt wurde, hat man gesehen, dass Petri-Netze Marken besitzen können. Kommen mehrere Marken in einem Petri-Netz vor, können diese bei einem simplen Petri-Netz nicht voneinander unterschieden werden. Die Bedeutung der Marke ist durch den modellierten Vorgang vorgegeben. Dadurch wird die Modellierung unterschiedlicher Ressourcen mit Marken in einem Petri-Netz schwierig. Eine Möglichkeit wie man den Fluss einer Ressource mit Hilfe eines Petri-Netzes und somit auch als Repräsentation in der PNML zeigen kann, ist mit den sogenannten *Workflow-Netzen* möglich.[Kin04] Workflow-Netze unterscheiden sich von den vorgestellten Stellen/Transitions-Netzen durch genau eine Startstelle und eine Endstelle, die Beginn und Ende eines Workflows sind.

Eine weitere Möglichkeit Ressourcen in Petri-Netzen zu modellieren ist die textuelle Beschreibung in Form einer Inschrift an der betreffenden Marke. Mit dem nachfolgenden Beispiel soll dies veranschaulicht werden.

Als Beispiel geht man von einem Amt aus, welches Anträge bearbeitet. Die Sachbearbeiter können die Anträge ablehnen oder genehmigen. Es gibt zwei verschiedene Arten von Anträgen: „Bauantrag“ und „Gewerbeantrag“. Wie schon erwähnt kann man mit dem einfachen Petri-Netz formal die zwei Arten von Anträgen nicht modellieren. Man kann allerdings die jeweiligen Netzelemente eindeutig beschriften oder aber man verwendet anstatt der Marken *Text-*

[2] http://www.thaiopensource.com/trex/
[3] http://relaxng.org/

marken, die die konkrete Ressource direkt beschreiben. *Textmarken* sind also nicht die, für Marken gewöhnlich verwendeten, schwarzen Punkte, sondern der beschreibende Text ist die Marke selbst. Im Beispiel für die Antragsbearbeitung (Abbildung 3) sind dies die zwei farblich hervorgehobenen Elemente „Bauantrag“ und „Gewerbeantrag“.

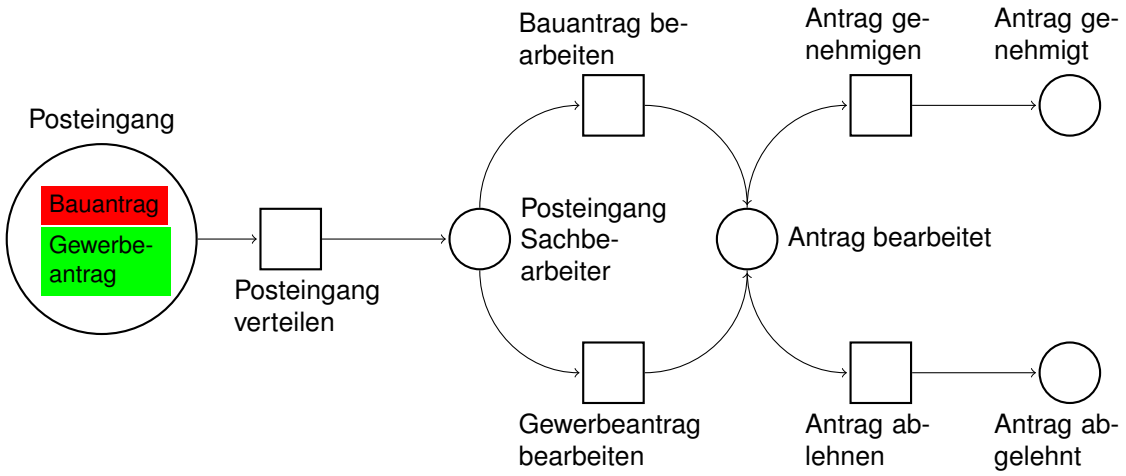

Abbildung 3: Bearbeitung von Bau- und Gewerbeanträgen als Petri-Netz

Will man eine Textmarke mithilfe der PNML darstellen, erfordert das zunächst die Definition eines Elements, welches diese *Textmarken* abbildet. Wie man in Listing 1 sehen kann, wird ein neuer Name für das entsprechende XML-Element, wie in Zeile 3 durch den Ausdruck `<element name="textMark">`, festgelegt. Als XML-Datentyp wurde ein String ausgewählt, da es sich um ein rein zeichenbasiertes Element handelt, welches nicht als numerischer Wert betrachtet werden muss.

```
1 <define name="TextuellMark"
2     xmlns:pnml="http://www.pnml.org/version-2009/grammar/pnml">
3   <element name="textMark">
4     <interleave>
5       <element name="text">
6         <data type="string"
7           datatypeLibrary="http://www.w3.org/2001/XMLSchema-
    datatypes"/>
8       </element>
9       <ref name="pnml:StandardAnnotationContent"/>
10    </interleave>
11  </element>
12 </define>
```

Listing 1: Definition einer Textmarke

Die Repräsentation der *Textmarken* in PNML könnte wie in dem Listing 2 aussehen. Dieses Beispiel zeigt vereinfacht die Repräsentation der *Stelle* „Posteingang“ aus dem Beispiel in Abbildung 3. Eine *Textmarke* wird wie in Zeile 15 zu sehen ist, mit `<textMark>` erstellt. Dazu

kommt die Bezeichnung, im Beispiel „Bauantrag“ und „Gewerbeantrag“ sowie eine Farbe, mit der die Bezeichnung hinterlegt werden soll. Die <graphics>-Elemente müssen für unsere *Textmarke* nicht extra definiert werden, da diese von den Standard-Annotationen der PNML geerbt werden. [BCH+03]

```
<pnml xmlns="http://www.example-namespace.com/">
  <net id="exampleNet1" type="http://www.example-netype.com/">
    ...
    <place id="s1">
      <graphics>
        <position x="x-coordinate" y="y-coordinate"/>
      </graphics>
      <name>
        <text>Posteingang</text>
        <graphics>
          <offset x="x-coordinate" y="y-coordinate"/>
        </graphics>
      </name>
      <textMark>
        <text>Bauantrag</text>
        <graphics>
          <fill color="red"/>
        </graphics>
      </textMark>
      <textMark>
        <text>Gewerbeantrag</text>
        <graphics>
          <fill color="green"/>
        </graphics>
      </textMark>
    </place>
    ...
  </net>
</pnml>
```

Listing 2: Repräsentation einer Textmarke in PNML

Mit den *Textmarken* hat man zwar ein anschauliches Mittel um verschiedene Ressourcen in ein und dem gleichen Petri-Netz darzustellen, allerdings sind diese Vorteile schon beim Thema Simulation und formale Darstellung wieder überholt. Es würden sich dann Fragen stellen, wie man diesen reinen Text formal darstellt, wie man damit eine sinnvolle automatisierte Analyse durchführen kann, wenn man doch gar nicht weiß was sich hinter der Beschriftung verbirgt.

Außerdem muss man anmerken, dass die Abbildungen 3 nicht ganz richtig ist. Die zwei *Textmarken*, die sich innerhalb der *Stelle* befinden, können bei einem Import in ein anderes Werkzeug auch außerhalb, also am Rand, einer *Stelle* erscheinen. Es ist dem Werkzeug überlassen,

wie es den weiteren Umgang, insbesonders mit der Position, mit unserem <**textMark**>-Element gestaltet. Die *ISO/IEC 15909-2* schlägt vor, dass dies in einer werkzeugspezifischen Information geregelt wird, damit kein Zwang für einzelne Werkzeuge entsteht. [HKK+09] Mit den *Textmarken* hat man eine Möglichkeit gesehen, wie man durch das Konzept der PNML neue Erweiterungen von Petri-Netzen abbilden kann. Ob und wie Erweiterungen genutzt werden, bleibt am Ende dem Anwender überlassen.

Neben der Abbildung von Ressourcen ist es möglich, dass in der Praxis der Fluss von bestimmten Ressourcen durch das Petri-Netz abgebildet werden soll. Auch hier kann man einen simplen Ansatz, ähnlich zu dem der *Textmarken*, wählen. Man teilt einer bestimmten Ressource eine Farbe zu und der Weg, welchen die Ressource von einem Knoten zu einem anderen Knoten durchlaufen hat, wird mit der entsprechenden Farbe markiert. In PNML realisiert man dies durch eine Erweiterung des <**arc**>-Elements mit den entsprechenden <**graphics**>-Elementen. Soll die gewünschte Kante zum Beispiel grün gefärbt werden so erreicht man das, durch den folgenden Ausdruck.

```
<arc id="k1" source="Quelle" target="Senke">
  <graphics>
    <position x="x-coordinate" y="y-coordinate"/>
    <line color="green" style="solid">
  </graphics>
</arc>
```

Das Attribut **color**="green" färbt die Kante grün und **style**="solid" zeichnet eine durchgehende Linie. Möglich sind aber auch gestrichelte und gepunkte Kanten. Dies verleiht weitere Gestaltungsspielräume bei der Abbildung von Ressourcenflüssen im Modell. Durch den simplen Ansatz ergeben sich aber auch bei dieser Erweiterungsmöglichkeit Probleme. Wie schon bei den *Textmarken*, sind auch bei dem Färben von Kanten, die Bedeutungen der Farben nicht formal definiert. Für ein einfaches Veranschaulichen des Umgangs mit verschiedenen Ressourcen in Petri-Netzen ist das aber ausreichend.

Bisher nicht betrachtet wurden besondere Schaltregeln, wie sie in höheren Petri-Netzen definiert sein können. Diese Schaltregeln können ebenfalls mit der PNML abgebildet werden. Hierzu werden spezielle Elemente eingeführt. Möchte man beispielsweise eine logische Konjunktion (*Und*) abbilden, benutzt man dazu einfach das Element <**and**>. Eine umfassende Übersicht findet sich in [HKK+09, Table 6]. Das Wichtige dabei ist, dass man auf die Verwendung der richtigen Petri Net Type Definition achtet.

4 Beispiel

Nach der Vorstellung der Erweiterungsmöglichkeiten im vorherigen Abschnitt, werden in diesem Abschnitt die vorgestellten Konzepte der Petri Net Markup Language und das Mapping auf XML anhand eines Prozessbeispiels veranschaulicht.

4.1 Beschreibung des Prozesses

Zuerst soll mit einer Beschreibung des Sachverhalts, der unserem Prozess zugrunde liegt, begonnen werden. Es geht dabei um die Planung und Durchführung einer Abteilungssitzung.

Zu Beginn erstellt der Abteilungsleiter eine Aufstellung der einzelnen Themen, die in der Sitzung besprochen werden sollen. Er greift dazu neben seinen eigenen Themen auch auf gewünschte Themen der anderen Teilnehmer zurück. Diese haben zuvor ihre Wunschthemen in eine Sitzungsdatenbank eingetragen.

Steht eine Auswahl der Themen fest, fragt der Abteilungsleiter einen Besprechungsraum an und wartet danach auf eine Reservierungsbestätigung. Nachdem der Abteilungsleiter die Reservierungsbestätigung erhalten hat, kann die Sitzung veröffentlicht werden, indem die Sitzung in die Sitzungsdatenbank eingetragen wird und parallel alle Teilnehmer per E-Mail eingeladen werden. Bevor die Sitzung beginnt erstellt der Abteilungsleiter eine Präsentation mit allen Sitzungsthemen. Wenn die Präsentation erstellt wurde, muss der Schlüssel des Besprechungsraums abgeholt werden. Kurz vor der Besprechung werden alle Teilnehmer nochmals an die Sitzung erinnert und der Besprechungsraum wird vorbereitet.

Während der Sitzung wird ein Protokoll über den Sitzungsverlauf angefertigt. Das Protokoll wird nach der Sitzung an alle Sitzungsteilnehmer verschickt und in der Sitzungsdatenbank abgelegt. Außerdem ist der Besprechungsraum nach der Sitzung wieder frei und der Raumschlüssel kann zurückgebracht werden.

4.2 Modellierung mit einem Petri-Netz

Bevor man das Mapping auf XML durchführt, modelliert man den gerade beschriebenen Sachverhalt als einfaches Petri-Netz mit *Stellen* und *Transitionen*. Abbildung 4 zeigt das vollständige Petri-Netz des Prozesses der Sitzungsorganisation als bipartiten Graphen. Wie man sieht besteht das Netz aus der 15-elementigen Menge S und einer Menge T mit zehn Elementen. 25 Elemente beinhaltet die Menge der *Flussrelationen* F. Auf eine Anfangsmarkierung wurde verzichtet, da dieses Modell lediglich die Umsetzung eines Petri-Netzes in die PNML zeigen soll und die Abbildung von Marken bereits vorgestellt wurde.

Zum Weiterverarbeiten des Petri-Netzes kann es mit einem beliebigen Werkzeug, das die PNML beherrscht, modelliert werden. Im vorliegenden Fall wurde das Werkzeug WoPeD[4] benutzt. Es bietet die Möglichkeit Petri-Netze im PNML-Format zu speichern und einzulesen. Außerdem ist es mit WoPeD möglich Simulationen mit einem Petri-Netz durchzuführen, was im vorliegenden Anwendungsfall allerdings nicht benötigt wird und für das Erstellen von PNML-Dateien nicht relevant ist.

4.3 Mapping auf XML

Das Petri-Netz der Sitzungsorganisation wurde mit dem vorgestellten Werkzeug modelliert und anschließend als Datei im PNML-Format gespeichert. Den Quelltext der Datei findet man im

[4] `http://woped.dhbw-karlsruhe.de/woped/`

Anhang B. Es ist anzumerken, dass der Übersichtlichkeit wegen werkzeugspezifische Elemente aus dem Quelltext gekürzt wurden. Dies betrifft im wesentlichen die <**transition**>-Elemente und die <**arc**>-Elemente. WoPeD stellt für diese Elemente erweiterte Möglichkeiten zur Verfügung und bildet diese in der PNML-Datei mit den <**toolspecific**>-Element ab. Im vorliegenden Fall muss man diesen Besonderheiten keine weitere Beachtung schenken.

Die gekürzte Datei kann aber von dem Werkzeug ohne weitere Probleme eingelesen und dargestellt werden, da bei einer korrekten Implementierung werkzeugspezifische Elemente nicht zwingend vorhanden sein müssen. Was bereits näher erläutert wurde.

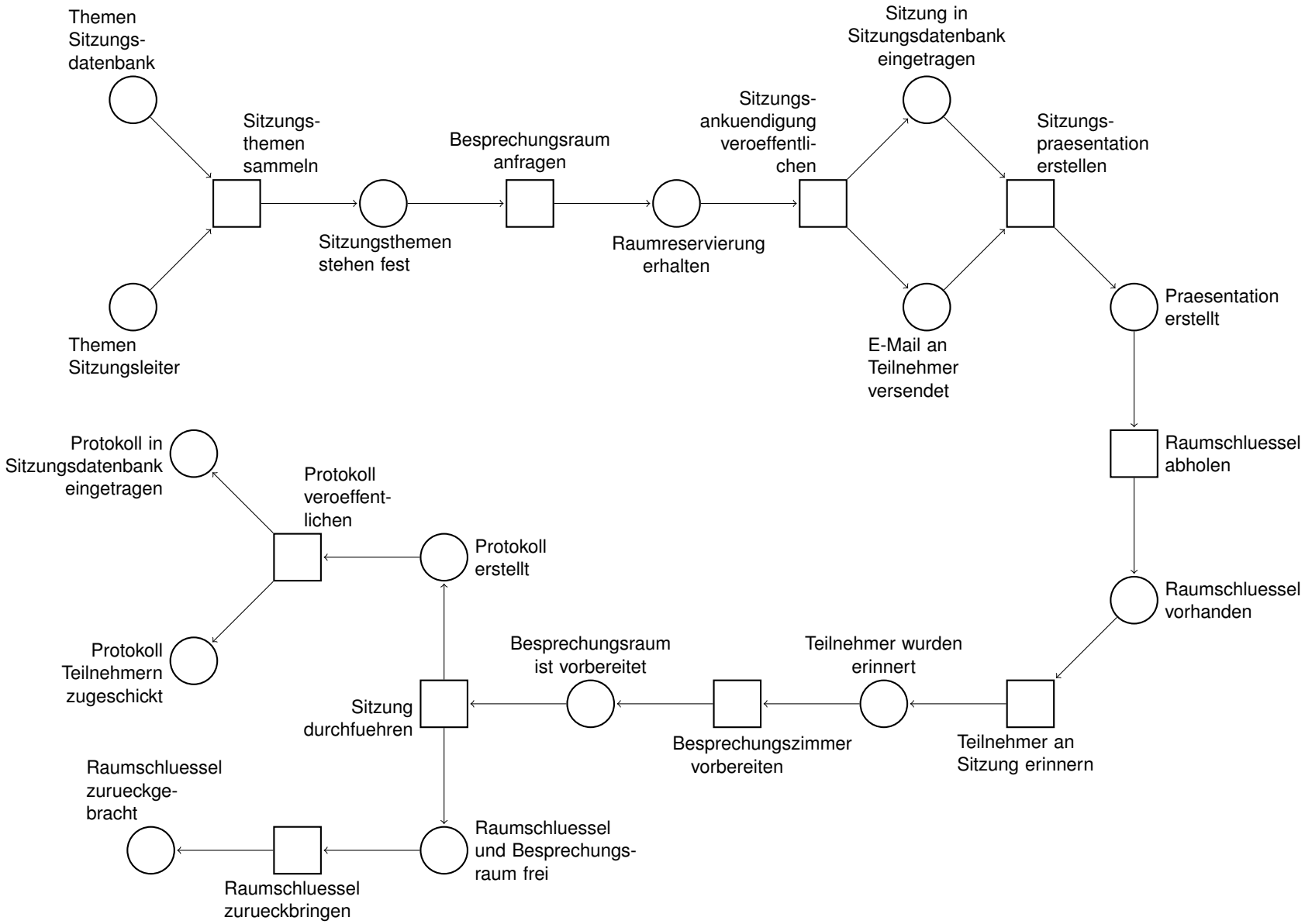

Abbildung 4: Petri-Netz einer Sitzungsorganisation

5 Fazit

Nach einer kurzen formalen Einführung in die wichtigsten Definitionen eines Petri-Netzes, wurden die wesentlichen Anforderungen an ein Austauschformat für Petri-Netze erläutert. Aus diesen Anforderungen heraus wurde aufgezeigt wie sich das Konzept der Petri Net Markup Language (PNML) entwickelt hat. Es wurde unter anderem deutlich, wie flexibel das Konzept in Hinsicht auf die Vielzahl an möglichen verschiedenen Petri-Netz-Typen gestaltet wurde. Ebenfalls wurden die einzelnen Elemente der PNML und deren Mapping auf XML-Elemente dargestellt. Hierbei zeigte sich auch, wie durch den modularen Aufbau des Austauschformats noch nicht vorhandene Petri-Netz-Typen gehandhabt werden können.

Ressourcen und Datenflüsse können mit der PNML durch Inschriften zumindest graphisch dargestellt werden. Allerdings konnte man auch sehen, dass eine formale Beschreibung zumindest mit den einfachen Petri-Netzen zu Problemen führen kann. Höhere Petri-Netze können diese Probleme lösen, indem sie entsprechende Formalismen bereitstellen und im Modell der PNML ausdrücklich vorgesehen sind. Die Unterscheidung zwischen einfachen und höheren Petri-Netzen wird in der PNML mit „basic“ und „structured“ PNML konsequent fortgesetzt.

Es wird deutlich, dass die PNML von Anfang an als erweiterbares Austauschformat entwickelt wurde und auch Möglichkeiten bietet, nicht etablierte und vom Standard weit entfernte Erweiterungen, wie die *Textmarken* aus Abschnitt 3.4 es sind, einzubeziehen.

Insgesamt wird die Petri Net Markup Language den an sie gestellten Anforderungen gerecht und setzt mit XML auf eine weit verbreitete und sehr flexible Auszeichnungssprache zur Umsetzung des Sprachmodells.

A Literaturverzeichnis

[AM95] Marco Ajmone Marsan (ed.), *Modelling with generalized stochastic Petri nets*, Wiley Series In Parallel Computing, Wiley, Chichester, 1995.

[BBK+00] Remi Bastide, Jonathan Billingtion, Ekkart Kindler, Fabrice Kordon, and Kjeld H. Mortensen, editors, *Meeting on XML/SGML based Interchange Formats for Petri Nets*, 21st ICATPN, June 2000.

[BCH+03] Jonathan Billington, Søren Christensen, Kees Hee, Ekkart Kindler, Olaf Kummer, Laure Petrucci, Reinier Post, Christian Stehno, and Michael Weber, *The Petri Net Markup Language: Concepts, Technology, and Tools*, Applications and Theory of Petri Nets 2003 (WilM.P. Aalst and Eike Best, eds.), Lecture Notes in Computer Science, vol. 2679, Springer, 2003, pp. 483–505.

[HKK+09] L. M. Hillah, Ekkart Kindler, F. Kordon, L. Petrucci, and N Tréves, *A primer on the Petri Net Markup Language and ISO/IEC 15909-2*, Petri Net Newsletter (2009), no. 76, 9–28.

[JK09] Kurt Jensen and Lars M. Kristensen, *Coloured Petri Nets*, Springer, Berlin, 2009.

[Kin04] Ekkhart Kindler, *Using the Petri Net Markup Language for Exchanging Business processes - Potential and Limitations*, XML4BPM 2004, XML Interchange Formats for Business Process Management, March 2004.

[KKB05] Uwe Kastens and Hans Kleine Büning, *Modellierung : Grundlagen und formale Methoden*, Hanser, München, 2005.

[Pet62] Carl Adam Petri, *Kommunikation mit Automaten*, Schriften des Institutes für Instrumentelle Mathematik, Bonn, 1962.

[Rei86] Wolfgang Reisig, *Petrinetze : eine Einführung*, 2., überarb. u. erw. aufl. ed., Studienreihe Informatik, Springer, Berlin, 1986.

[WK03] Michael Weber and Ekkart Kindler, *The Petri Net Markup Language*, Petri Net Technology for Communication-Based Systems (Hartmut Ehrig, Wolfgang Reisig, Grzegorz Rozenberg, and Herbert Weber, eds.), Lecture Notes in Computer Science, vol. 2472, Springer Berlin Heidelberg, 2003, pp. 124–144.

B Quelltext PNML-Datei des Beispiels

Listing 3: Repräsentation des Petri-Netzes aus Abb. 4 in PNML

```
<?xml version="1.0" encoding="UTF-8"?>
<!--PLEASE DO NOT EDIT THIS FILE
Created with Workflow PetriNet Designer Version 1.0 (woped.org)-->
<pnml>
  <net type="http://www.informatik.hu-berlin.de/top/pntd/ptNetb" id="noID
    ">
    <place id="p7">
      <name>
        <text>E-Mail an Teilnehmer versendet</text>
        <graphics>
          <offset x="750" y="270"/>
        </graphics>
      </name>
      <graphics>
        <position x="750" y="230"/>
        <dimension x="40" y="40"/>
      </graphics>
    </place>
    <place id="p6">
      <name>
        <text>Sitzung in Sitzungsdatenbank eingetragen</text>
        <graphics>
          <offset x="750" y="110"/>
        </graphics>
      </name>
      <graphics>
        <position x="750" y="70"/>
        <dimension x="40" y="40"/>
      </graphics>
    </place>
    <place id="p5">
      <name>
        <text>Raumreservierung erhalten</text>
        <graphics>
          <offset x="520" y="180"/>
        </graphics>
      </name>
      <graphics>
        <position x="520" y="140"/>
        <dimension x="40" y="40"/>
      </graphics>
    </place>
    <place id="p4">
      <name>
        <text>Sitzungsthemen stehen fest</text>
        <graphics>
          <offset x="310" y="180"/>
        </graphics>
      </name>
      <graphics>
        <position x="310" y="140"/>
        <dimension x="40" y="40"/>
      </graphics>
    </place>
    <place id="p3">
```

```
55        <name>
            <text>Themen Sitzungsleiter</text>
            <graphics>
              <offset x="60" y="240"/>
            </graphics>
60        </name>
          <graphics>
            <position x="60" y="200"/>
            <dimension x="40" y="40"/>
          </graphics>
65      </place>
        <place id="p2">
          <name>
            <text>Themen Sitzungsdatenbank</text>
            <graphics>
70            <offset x="70" y="110"/>
            </graphics>
          </name>
          <graphics>
            <position x="70" y="70"/>
75          <dimension x="40" y="40"/>
          </graphics>
        </place>
        <place id="p9">
          <name>
80          <text>Raumschluessel vorhanden</text>
            <graphics>
              <offset x="1010" y="490"/>
            </graphics>
          </name>
85        <graphics>
            <position x="1010" y="450"/>
            <dimension x="40" y="40"/>
          </graphics>
        </place>
90      <place id="p8">
          <name>
            <text>Praesentation erstellt</text>
            <graphics>
              <offset x="1010" y="190"/>
95          </graphics>
          </name>
          <graphics>
            <position x="1010" y="150"/>
            <dimension x="40" y="40"/>
100       </graphics>
        </place>
        <place id="p10">
          <name>
            <text>Teilnehmer wurden erinnert</text>
105         <graphics>
              <offset x="720" y="490"/>
            </graphics>
          </name>
          <graphics>
110         <position x="720" y="450"/>
            <dimension x="40" y="40"/>
          </graphics>
        </place>
        <place id="p16">
```

```
115         <name>
              <text>Raumschluessel zurueckgebracht</text>
              <graphics>
                <offset x="60" y="580"/>
              </graphics>
120         </name>
            <graphics>
              <position x="60" y="540"/>
              <dimension x="40" y="40"/>
            </graphics>
125       </place>
          <place id="p15">
            <name>
              <text>Protokoll Teilnehmern zugeschickt</text>
              <graphics>
130             <offset x="60" y="470"/>
              </graphics>
            </name>
            <graphics>
              <position x="60" y="430"/>
135           <dimension x="40" y="40"/>
            </graphics>
          </place>
          <place id="p14">
            <name>
140           <text>Protokoll in Sitzungsdatenbank eingetragen</text>
              <graphics>
                <offset x="60" y="330"/>
              </graphics>
            </name>
145         <graphics>
              <position x="60" y="290"/>
              <dimension x="40" y="40"/>
            </graphics>
          </place>
150       <place id="p13">
            <name>
              <text>Raumschluessel und Besprechungsraum frei</text>
              <graphics>
                <offset x="300" y="580"/>
155           </graphics>
            </name>
            <graphics>
              <position x="300" y="540"/>
              <dimension x="40" y="40"/>
160         </graphics>
          </place>
          <place id="p12">
            <name>
              <text>Protokoll erstellt</text>
165           <graphics>
                <offset x="300" y="400"/>
              </graphics>
            </name>
            <graphics>
170           <position x="300" y="360"/>
              <dimension x="40" y="40"/>
            </graphics>
          </place>
          <place id="p11">
```

```
175         <name>
              <text>Besprechungsraum ist vorbereitet</text>
              <graphics>
                <offset x="470" y="490"/>
              </graphics>
180         </name>
            <graphics>
              <position x="470" y="450"/>
              <dimension x="40" y="40"/>
            </graphics>
185       </place>
          <transition id="t3">
            <name>
              <text>Besprechungsraum anfragen</text>
              <graphics>
190             <offset x="420" y="180"/>
              </graphics>
            </name>
            <graphics>
              <position x="420" y="140"/>
195           <dimension x="40" y="40"/>
            </graphics>
          </transition>
          <transition id="t2">
            <name>
200           <text>Sitzungsthemen sammeln</text>
              <graphics>
                <offset x="200" y="180"/>
              </graphics>
            </name>
205         <graphics>
              <position x="200" y="140"/>
              <dimension x="40" y="40"/>
            </graphics>
          </transition>
210       <transition id="t10">
            <name>
              <text>Protokoll veroeffentlichen</text>
              <graphics>
                <offset x="190" y="400"/>
215           </graphics>
            </name>
            <graphics>
              <position x="190" y="360"/>
              <dimension x="40" y="40"/>
220         </graphics>
          </transition>
          <transition id="t11">
            <name>
              <text>Raumschluessel zurueckbringen</text>
225           <graphics>
                <offset x="190" y="580"/>
              </graphics>
            </name>
            <graphics>
230           <position x="190" y="540"/>
              <dimension x="40" y="40"/>
            </graphics>
          </transition>
          <transition id="t4">
```

```
235         <name>
              <text>Sitzungsankuendigung veroeffentlichen</text>
              <graphics>
                <offset x="620" y="180"/>
              </graphics>
240         </name>
            <graphics>
              <position x="620" y="140"/>
              <dimension x="40" y="40"/>
            </graphics>
245       </transition>
          <transition id="t5">
            <name>
              <text>Sitzungspraesentation erstellen</text>
              <graphics>
250             <offset x="880" y="190"/>
              </graphics>
            </name>
            <graphics>
              <position x="880" y="150"/>
255           <dimension x="40" y="40"/>
            </graphics>
          </transition>
          <transition id="t6">
            <name>
260           <text>Raumschluessel abholen</text>
              <graphics>
                <offset x="1010" y="320"/>
              </graphics>
            </name>
265         <graphics>
              <position x="1010" y="280"/>
              <dimension x="40" y="40"/>
            </graphics>
          </transition>
270       <transition id="t7">
            <name>
              <text>Teilnehmer an Sitzung erinnern</text>
              <graphics>
                <offset x="850" y="490"/>
275           </graphics>
            </name>
            <graphics>
              <position x="850" y="450"/>
              <dimension x="40" y="40"/>
280         </graphics>
          </transition>
          <transition id="t8">
            <name>
              <text>Besprechungszimmer vorbereiten</text>
285           <graphics>
                <offset x="590" y="490"/>
              </graphics>
            </name>
            <graphics>
290           <position x="590" y="450"/>
              <dimension x="40" y="40"/>
            </graphics>
          </transition>
          <transition id="t9">
```

```
295         <name>
              <text>Sitzung durchfuehren</text>
              <graphics>
                <offset x="370" y="490"/>
              </graphics>
300         </name>
            <graphics>
              <position x="370" y="450"/>
              <dimension x="40" y="40"/>
            </graphics>
305       </transition>
          <arc id="a20" source="t8" target="p11">
            <inscription>
              <text>1</text>
            </inscription>
310         <graphics/>
          </arc>
          <arc id="a9" source="p5" target="t4">
            <inscription>
              <text>1</text>
315         </inscription>
            <graphics/>
          </arc>
          <arc id="a16" source="t6" target="p9">
            <inscription>
320           <text>1</text>
            </inscription>
            <graphics/>
          </arc>
          <arc id="a17" source="p9" target="t7">
325         <inscription>
              <text>1</text>
            </inscription>
            <graphics/>
          </arc>
330       <arc id="a14" source="t5" target="p8">
            <inscription>
              <text>1</text>
            </inscription>
            <graphics/>
335       </arc>
          <arc id="a15" source="p8" target="t6">
            <inscription>
              <text>1</text>
            </inscription>
340         <graphics/>
          </arc>
          <arc id="a12" source="p7" target="t5">
            <inscription>
              <text>1</text>
345         </inscription>
            <graphics/>
          </arc>
          <arc id="a13" source="p6" target="t5">
            <inscription>
350           <text>1</text>
            </inscription>
            <graphics/>
          </arc>
          <arc id="a10" source="t4" target="p6">
```

```
355         <inscription>
              <text>1</text>
            </inscription>
            <graphics/>
          </arc>
360       <arc id="a11" source="t4" target="p7">
            <inscription>
              <text>1</text>
            </inscription>
            <graphics/>
365       </arc>
          <arc id="a3" source="p2" target="t2">
            <inscription>
              <text>1</text>
            </inscription>
370         <graphics/>
          </arc>
          <arc id="a5" source="p3" target="t2">
            <inscription>
              <text>1</text>
375         </inscription>
            <graphics/>
          </arc>
          <arc id="a6" source="t2" target="p4">
            <inscription>
380           <text>1</text>
            </inscription>
            <graphics/>
          </arc>
          <arc id="a7" source="p4" target="t3">
385         <inscription>
              <text>1</text>
            </inscription>
            <graphics/>
          </arc>
390       <arc id="a18" source="t7" target="p10">
            <inscription>
              <text>1</text>
            </inscription>
            <graphics/>
395       </arc>
          <arc id="a8" source="t3" target="p5">
            <inscription>
              <text>1</text>
            </inscription>
400         <graphics/>
          </arc>
          <arc id="a19" source="p10" target="t8">
            <inscription>
              <text>1</text>
405         </inscription>
            <graphics/>
          </arc>
          <arc id="a25" source="t11" target="p16">
            <inscription>
410           <text>1</text>
            </inscription>
            <graphics/>
          </arc>
          <arc id="a26" source="p12" target="t10">
```

```
415         <inscription>
              <text>1</text>
            </inscription>
            <graphics/>
          </arc>
420       <arc id="a27" source="t10" target="p15">
            <inscription>
              <text>1</text>
            </inscription>
            <graphics/>
425       </arc>
          <arc id="a28" source="t10" target="p14">
            <inscription>
              <text>1</text>
            </inscription>
430         <graphics/>
          </arc>
          <arc id="a21" source="p11" target="t9">
            <inscription>
              <text>1</text>
435         </inscription>
            <graphics/>
          </arc>
          <arc id="a22" source="t9" target="p12">
            <inscription>
440           <text>1</text>
            </inscription>
            <graphics/>
          </arc>
          <arc id="a23" source="t9" target="p13">
445         <inscription>
              <text>1</text>
            </inscription>
            <graphics/>
          </arc>
450       <arc id="a24" source="p13" target="t11">
            <inscription>
              <text>1</text>
            </inscription>
            <graphics/>
455       </arc>
        </net>
    </pnml>
```

C Eidestattliche Erklärung

Ich versichere hiermit wahrheitsgemäß, die Arbeit selbständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderung entnommen wurde.

Ort, Datum

Unterschrift