

Semantische Annotation von Petri-Netzen

Agnes Koschmider und Daniel Ried

Institut für Angewandte Informatik und Formale Beschreibungsverfahren
Universität Karlsruhe (TH), 76187 Karlsruhe, Germany
{koschmider, ried}@aifb.uni-karlsruhe.de

1 Einführung

Durch unternehmensübergreifende Kooperationen können Abläufe rationalisiert, Synergien zwischen Geschäftspartnern gebildet und Unternehmensrisiken gemindert werden. Dies erfordert flexibel anpassbarer Geschäftsprozesse, die schnell mit Prozessen von Unternehmenspartnern zusammengeführt werden können. Voraussetzung für eine effiziente Prozesszusammenführung sind maschinenlesbare und maschineninterpretierbare Prozesskomponenten. Bei der Modellierung von Geschäftsprozessen in unterschiedlichen Unternehmen kann ein einheitliches Vokabular nicht vorausgesetzt werden, weshalb Prozessdaten von Systemen unterschiedlich verstanden und interpretiert werden. Durch die Verwendung von Ontologien kann ein einheitliches Verständnis über Begriffe und Beziehungen von Geschäftsprozessen definiert und die Wiederverwendung von Geschäftsprozess-Komponenten und Interoperabilität in ungleichen Systemlandschaften unterstützt werden. Zur Modellierung, Simulation und Validierung von Geschäftsprozessen eignen sich Petri-Netze, die sowohl eine formale als auch eine graphische Darstellung von Petri-Netz-Elementen ermöglichen.

In diesem Beitrag werden wir eine Ontologie zur semantischen Annotation von Petri-Netzen vorstellen. Die Ontologie wird mit der Web Ontology Language (OWL) [W3C04a] definiert, die seit Anfang 2004 als Empfehlung vom W3C veröffentlicht wurde. OWL gibt es in den Ausprägungen OWL Lite, OWL DL und OWL Full, wobei gilt: $OWL\ Lite \subset OWL\ DL \subset OWL\ Full$ [W3C04b]. OWL DL (Description Logic) ist eine syntaktische Variante der $SHOIN(D)$ Beschreibungslogik und ist trotz ihres hohen Grads an Ausdrucksmächtigkeit entscheidbar [Hor05]. $SHOIN(D)$ bietet Verneinung, Disjunktion und eine eingeschränkte Form von All- und Existenzquantoren für Variablen. In Kombination mit bestimmten Aussagen erlauben Quantoren das automatische Schlussfolgern in Bezug auf andere Aussagen. Durch die Verwendung von Schlussfolgerungsmechanismen in semantisch annotierten Geschäftsprozessen könnte beispielsweise die Prozesszusammenführung von unternehmensübergreifenden Kooperationen durch Software implementierte Komponenten unterstützt werden. Ziel unserer Forschungsarbeit sind Flexibilisierung, leichtere Integration und automatische Zusammenführung von Geschäftsprozessen, auch wenn Geschäftspartner ein unterschiedliches Vokabular verwenden.

Unser Beitrag ist wie folgt gegliedert: zunächst werden wir eine Ontologie für elementare Petri-Netze definieren und diese Ontologie in Abschnitt 3 um ontologische Konzepte für Prädikate/Transitionen-Netze erweitern. Abschnitt 4 stellt SemPeT vor, ein Werkzeug zur Modellierung von Prädikate/Transitionen-Netzen und semantischen Annotation der Modelle durch OWL. Unser Beitrag endet mit einer Zusammenfassung und einem Ausblick auf zukünftige Forschungsarbeiten.

2 Ontologie für Petri-Netze

Ein Petri-Netz [RR98] ist ein gerichteter bipartiter Graph mit Knoten und Kanten und wird durch das Tripel $N = (S, T, F)$ beschrieben. Knoten sind Stellen (S-Elemente) und Transitionen (T-Elemente), die durch Kanten (F-Elemente) verbunden sind, wobei folgenden Eigenschaften gelten: (i) $S \cap T = \emptyset$ und (ii) $F \subseteq (S \times T) \cup (T \times S)$. In Petri-Netzen unterscheiden wir zwei Arten von Kanten: $F_r \subseteq (S \times T)$

und $F_w \subseteq (T \times S)$. Zur Abbildung von Petri-Netz-Elementen auf ontologische Konzepte, verwenden wir die OWL-Grundelemente Klassen (Class), Objekteigenschaften (ObjectProperty), Instanzen (Individuum) und Datentypeneigenschaften (DatatypeProperty)¹. Petri-Netz-Elemente definieren wir in OWL wie folgt:

- **PetriNet** für alle möglichen Petri-Netze $N = (S, T, F)$
- **Place** für alle Stellen aus S ,
- **Transition** für alle Transitionen aus T ,
- **FromPlace** für alle Kanten aus F_r ,
- **ToPlace** für alle Kanten aus F_w .

Um auszudrücken, dass Petri-Netze Kanten und Knoten haben, ordnen wir der Klasse **PetriNet** die Objekteigenschaften

- **hasNode** für die Domäne **PetriNet** und den Wertebereich **Place** und **Transition**,
- **hasArc** für die Domäne **PetriNet** und den Wertebereich **FromPlace** und **ToPlace**

zu. Abhängig vom Petri-Netz-Typ beinhalten Stellen Marken. Diese Eigenschaft drücken wir durch die OWL-Objekteigenschaft

- **hasMarking** für die Domäne **Place**

aus. Petri-Netze folgen einer operationalen Semantik, die die Zusammenführung und Verifikation von Geschäftsprozessen ermöglicht. Um dies in OWL auszudrücken, verwenden wir die folgenden Objekteigenschaften:

- **placeRef** für die Domäne **Transition** und den Wertebereich **Place**,
- **transRef** für die Domäne **Place** und den Wertebereich **Transition**.

Im Folgenden werden wir die Petri-Netz-Ontologie um Elemente für strikte Prädikate/Transitionen-Netze erweitern.

3 Ontologie für Pr/T-Netze

Prädikate/Transitionen-Netze (Pr/T-Netze) sind eine Variante von höheren Petri-Netzen [GL81]. Im Gegensatz zu elementaren Petri-Netzen (S/T-Netzen), in denen Marken anonyme Objekte darstellen, repräsentieren Marken in höheren Petri-Netzen Objekte mit individuellen Eigenschaften. Ein striktes Pr/T-Netz ist ein Tupel $PrT = (S, T, F, \Psi, KB, TI, M^0)$, das die folgenden Bedingungen erfüllt (i) (S, T, F) ist ein Netz und S wird als Menge von Prädikaten mit veränderlichen Ausprägungen und T als Menge von Transitionsschemata interpretiert. (ii) $\Psi = (D, FT, PR)$ ist eine Struktur bestehend aus einer Individuenmenge D , einer auf D definierten Menge von Funktionen FT und einer Menge PR von auf D definierten Prädikaten mit veränderlichen Ausprägungen. (iii) Die Kantenbeschriftung KB weist jeder Kante aus F eine Menge von Variablentupeln mit der Stelligkeit des adjazenten Prädikats zu. (iv) TI weist jeder Transition aus T eine Transitionsinschrift in Form eines über Ψ und der Menge der an den adjazenten Kanten vorkommenden Variablen gebildeten prädikatenlogischen Ausdrucks zu. (v) M^0 ist eine Markierung der Prädikate mit Mengen von konstanten Individuentupeln, deren Stelligkeit der Stelligkeit des Prädikats entspricht.

Um ein Pr/T-Netz in OWL auszudrücken erweitern wir die oben definierten OWL-Klassen und Objekteigenschaften um:

¹In der OWL-Spezifikation werden noch Eigenschaftsmerkmale, Eigenschaftsrestriktionen oder komplexe Klassen definiert.

- **LogicalConcept** für alle Transitionsinschriften aus TI ,
- **IndividualDataItem** für alle Prädikate PR ,
- **Attribute** für alle Funktionen aus FT ,
- **Value** für alle Teilmengen aus $D_1 \times \dots \times D_n$,
- **Delete, Insert** für die Kantenbeschriftung mit Mengen von Variablentupeln.

Die Annotation einer Transition eines Pr/T-Netzes setzt sich aus Schaltbedingungen und Schaltoperationen zusammen, die in einer prädikatenlogischen Formel vereinigt werden. Bedingungen und Operationen drücken wir wie folgt aus:

- **Condition** für die Schaltbedingungen,
- **Operation** für die Schaltoperationen.

Den Klassen weisen wir Objekteigenschaften wie folgt zu:

- **hasLogicalConcept** mit der Domäne **Transition** und dem Wertebereich **LogicalConcept**
- **hasOperation** mit der Domäne **LogicalConcept** und dem Wertebereich **Operation**,
- **hasCondition** mit der Domäne **LogicalConcept** und dem Wertebereich **Condition**,
- **hasInscription** mit der Domäne **FromPlace, ToPlace** und dem Wertebereich **Delete, Insert**.

Für die Klassen **LogicalConcept**, **IndividualDataItem**, **Delete** und **Insert** haben wir die Objekteigenschaften:

- **hasAttribute** mit Wertebereich **Attribute**,

und für **Attribute** die Objekteigenschaft

- **hasValue** mit dem Wertebereich **Value**

identifiziert. Für die Klasse **Value** gibt es die folgende Objekteigenschaft:

- **hasRef** mit der Domäne und dem Wertebereich **Value**.

Um eine prädikatenlogische Formel in der Pr/T-Ontologie als Schaltbedingung einer Transition darstellen zu können, muss diese in Pränex-Normalform vorliegen. Besitzt die Formel desweiteren eine Matrix in KNF, so kann diese durch die folgenden drei charakteristischen Datentypeigenschaften dargestellt werden:

- **forall, exists** und **and** mit der Domäne **Condition** und dem Wertebereich **xsd:string**

Eine Schaltoperation einer Transition wird durch die Datentypeigenschaft

- **function** mit der Domäne **Operation** und dem Wertebereich **xsd:string** wiedergegeben.

Die definierte Pr/T-Ontologie kann formal mit einer Beschreibungslogik abgebildet werden. In der oben definierten Ontologie haben wir nur Klassen, Objekt- und Datentypeneigenschaften definiert. Um allerdings ein Pr/T-Netz vollständig in OWL auszudrücken, benötigen wir noch die Konstrukte *owl:disjointWith* um Disjunktion zwischen **Transition** und **Place** auszudrücken und Quantoren um beispielsweise zu beschreiben, dass für alle Kanten **FromPlace** der Knoten **Place** angegeben werden muss und für **ToPlace** der Knoten **Transition**². OWL DL ist eine syntaktische Variante der *SHOIN(D)*³ Beschreibungslogik und ist trotz ihres hohen Grads an Ausdrucksmächtigkeit entscheidbar. In Schlussfolgerungssystemen wie Racer und FaCT wird die Beschreibungslogik *SHIQ(D)* [HST99] verwendet und unterscheidet sich von *SHOIN(D)* vor allem dadurch, dass keine Nominale unterstützt werden⁴. Um Entscheidbarkeit in *SHIQ(D)* zu erreichen, wird OWL DL um Regeln, die DL-sicher sind, erweitert [MSS04]. Abfragen für diese Erweiterung von *SHIQ(D)* mit DL-sicheren Regeln werden durch einen Algorithmus erreicht, der *SHIQ(D)* zu disjunktiven Datalog-Programmen reduziert [HMS04].

In Abbildung 1 wird die Ontologie für Pr/T-Netze in einem Klassendiagramm gezeigt:

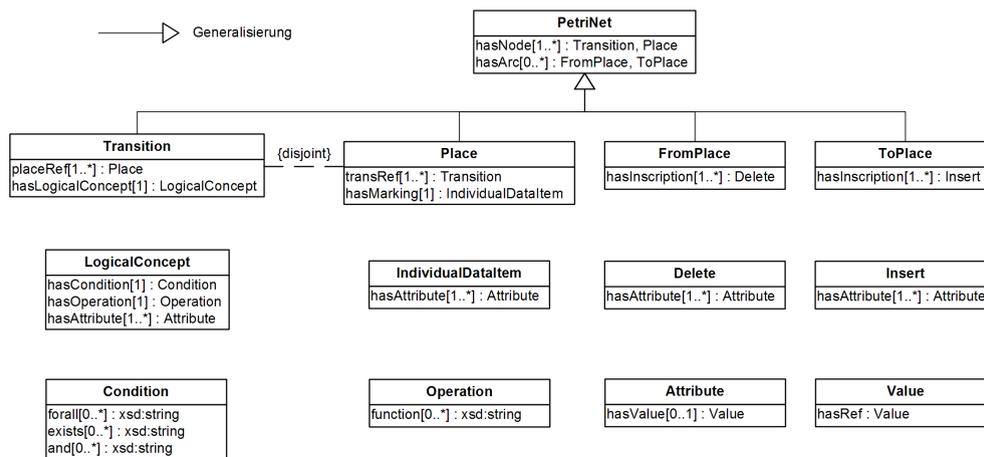


Figure 1: Klassendiagramm der Ontologie für Pr/T-Netze

4 Implementierung der Pr/T-Netz-Ontologie

Abbildung 1 zeigt das in den Abschnitten 2 und 3 definierte Vokabular der Ontologie. Eine Ontologiekategorie ist darin als Klasse dargestellt. Eine Objekt- bzw. Datentypeneigenschaft der Ontologie ist als Attribut der Klasse dargestellt, die als Domäne der Eigenschaft definiert ist. Der Datentyp und die Multiplizität eines Attributs sind als Wertebereich bzw. Kardinalität der Eigenschaft zu interpretieren. Liegt eine Ontologiekategorie in der Domäne bzw. im Wertebereich einer Eigenschaft, so gilt dieses ebenfalls für alle Unterklassen. Die Implementierung des Vokabulars erfolgt anhand der von Jena⁵ – einer Java API für RDF/OWL – zur Verfügung gestellten Klassen *Class* bzw. *ObjectProperty* und *DatatypeProperty* innerhalb einer sogenannten Vokabularklasse. Die taxonomischen Beziehungen der Ontologieklassen untereinander sowie diesbezügliche Einschränkungen, wie z.B. die Disjunktheit von Klassen, werden anhand eines Ontologiemodells (*OntModel*) implementiert. Durch diese Vorgehensweise können Java-Anwendungen auf die Ontologie sowie deren Vokabularelemente zugreifen und Instanzmodelle erzeugt werden.

Das im Rahmen dieses Beitrags vorgestellte Werkzeug *SemPeT* unterstützt die grafische Modellierung und semantische Annotation von Pr/T-Netz-Instanzen durch die in Abbildung 2 dargestellte GUI.

Als grundlegende interne Datenstruktur einer modellierten Instanz wird ein von Jena bereitgestelltes OWL-Modell verwendet. Beim Hinzufügen von grafischen Elementen bzw. bei der Definition und Initialisierung

²Der Vorgänger und Nachfolgeknoten können durch die Objekteigenschaften **placeRef** und **transitionRef** ermittelt werden

³*O* steht für *nominals*, *N* für *unqualified number restrictions* und *(D)* für *datatypes*

⁴Nominale in der OWL DL-Syntax werden durch *owl:oneOf* oder *owl:hasValue* ausgedrückt.

⁵<http://www.hpl.hp.com/semweb/jena.htm>

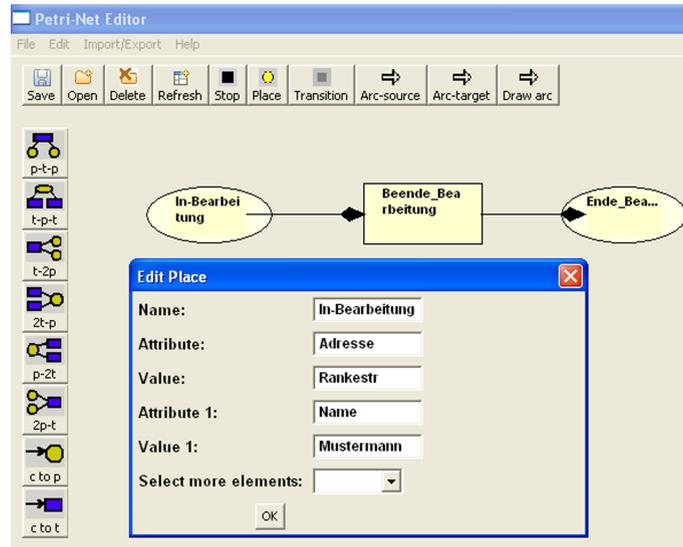


Figure 2: Grafische Modellierung von Pr/T-Netzen mit SemPeT

von Annotationen werden dem Ontologiemodell entsprechende, bzgl. der Ontologie gültige Aussagen aus dem Vokabular hinzugefügt. Eine Aussage ist ein Tripel (Subjekt, Prädikat, Objekt) mit einer Eigenschaft als Prädikat und einer Ontologiekategorie aus der Domäne der Eigenschaft als Subjekt sowie mit einer Klasse bzw. einem literalen Wert (z.B. vom Typ `xsd:string` oder `xsd:nonNegativeInteger`) als Objekt. Dabei wird gewährleistet, dass eine Pr/T-Netz-Instanz korrekt auf ein Instanzmodell der vorgestellten Ontologie innerhalb des definierten Vokabulars abgebildet wird.

Die Ausgabe des Modells erfolgt serialisiert in OWL-Syntax und hat die folgende Form (Ausschnitt):

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns="http://www.aifb.uni-karlsruhe.de/apparat#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:petri="http://www.aifb.uni-karlsruhe.de/petri/PNOntologie.owl#"
  xml:base="http://www.aifb.uni-karlsruhe.de/apparat">
  <petri:PetriNet rdf:ID="Arbeitsbearbeitung">
    <petri:hasNode>
      <petri:Place rdf:ID="In-Bearbeitung">
        <petri:hasMarking>
          <petri:IndividualDataItem rdf:ID="R_In-Bearbeitung">
            <petri:hasAttribute rdf:resource="#Adresse"/>
            <petri:hasAttribute rdf:resource="#Name"/>
          </petri:IndividualDataItem>
        </petri:hasMarking>
      </petri:Place>
      <petri:transRef>
        <petri:Transition rdf:ID="Beende_Bearbeitung">
          <petri:hasNode rdf:resource="#Beende_Bearbeitung"/>
        </petri:Transition>
      </petri:transRef>
    </petri:hasNode>
  </petri:PetriNet>
</rdf:RDF>
```

Auf diese Weise wird die maschinelle Weiterverarbeitung, d.h. der unternehmensübergreifende Austausch von Prozessmodellen, die Interpretation und Validierung von Prozessmodellen anhand eines einheitlichen

Vokabulars sowie das Ableiten von Schlussfolgerungen aus den Modellen anhand eines Regelsatzes unterstützt.

5 Zusammenfassung und Ausblick

In unserem Beitrag haben wir eine semantische Annotation für Pr/T-Netze vorgestellt, womit Daten von Pr/T-Netzen in einem maschinenverständlichen und maschineninterpretierbaren Format abgebildet werden können. Derzeit implementieren wir auch ein Werkzeug, das die semantische Annotation von Pr/T-Netzen unterstützt. Aufbauend auf der Definition einer Ontologie für Pr/T-Netze untersuchen wir Techniken des Ontology Alignment, die Grundlage eines Mappings einer Pr/T-Ontologie auf eine andere Pr/T-Ontologie bilden. Durch die semantische Zusammenführung von mehreren Pr/T-Ontologien ergibt sich auch die Grundlage für weitere interessante Fragestellungen. Es wird beispielsweise zu untersuchen sein, in wieweit Ontologien um Zeitaspekte erweitert werden können, damit die zusammengeführten Pr/T-Ontologien ihre Ausgangssemantik behalten.

Literaturverzeichnis

- [GL81] H. J. Genrich und K. Lautenbach. System modelling with high level Petri nets. *Theoretical Computer Science*, (13):109–136, 1981.
- [HMS04] U. Hustadt, B. Motik und U. Sattler. Reducing $SHIQ^-$ Description Logic to Disjunctive Datalog Programs. In *Proceedings of the 9th International Conference on Knowledge Representation and Reasoning*, Seiten 152–162. 2004.
- [Hor05] I. Horrocks. Applications of description logics: State of the art and research challenges. In *Proceeding of the 13th International Conference on Conceptual Structures*, Seiten 87–90. Springer, 2005.
- [HST99] I. Horrocks, U. Sattler und S. Tobies. Practical reasoning for expressive description logics. In *Proceeding of the 6th International Conference on Logic for Programming and Automated Reasoning*, Seiten 161–180. 1999.
- [MSS04] B. Motik, U. Sattler und R. Studer. Query Answering for OWL-DL with Rules. In *Proceeding of the 3rd International Semantic Web Conference*. 2004.
- [RR98] W. Reisig und G. Rozenberg. *Lectures on Petri Nets: Basic Models*, Band 1491 der *Lecture Notes in Computer Science*. Springer, 1. Auflage, 1998.
- [W3C04a] World Wide Web Consortium. *OWL Web Ontology Language Guide*, 10. Februar 2004. Recommendation. <http://www.w3.org/TR/2004/REC-owl-guide-20040210/>.
- [W3C04b] World Wide Web Consortium. *OWL Web Ontology Language Semantics and Abstract Syntax*, 10. Februar 2004. Recommendation. <http://www.w3.org/TR/2004/REC-owl-semantics-20040210/>.