Modelling Examination Regulations in OWL

PA Abschlussvortrag
am 14.12.2006
Alexander Pohl

Gliederung

- Einführung
  - Aufgaben und Ziele der Projektarbeit
  - Annahmen der offenen und abgeschlossenen Welt im Modell der Prüfungsordnung
- Unzulänglichkeiten bei der Modellierung und Lösungsvorschläge
  - Werte-Einschränkung von Datentypen
  - Arithmetik in OWL
  - „Finite Quantification“
  - „Qualified Number Restrictions“
  - Verwendung von SWRL
- Zusammenfassung

Einführung

Aufgaben und Ziele der Projektarbeit

- Modellierung eines einfachen Vertrags in OWL
  - → Sehr einfache Prüfungsordnung als Repräsentanten
- Identifizieren von Problemen bei der Modellierung in OWL durch
  - mangelnde Ausdrucksfähigkeit von OWL
  - syntaktische Unzulänglichkeiten von OWL
- Vorschlagen von Lösungen für die Probleme

Annahmen der offenen und abgeschlossenen Welt im Modell der Prüfungsordnung

- Annahme der abgeschlossenen Welt für das Schließen über Prüfungen ist sinnvoll
  - Annahme, daß eine Klausur nicht abgelegt wurde, wenn dieser kein Ergebnis zugeteilt wurde, ist berechtigt
- Annahme der offenen Welt für das Schließen über Studenten ist sinnvoll
  - Annahme, daß ein Student der Beste ist, nur weil er momentan der einzige Student in der Wissensbasis ist, ist unberechtigt

Unzulänglichkeiten bei der Modellierung und Lösungsvorschläge

Werte-Einschränkung von Datentypen

Problem:

Studentenklassen u.a. durch bestimmte Wertebereiche von Attributen festgelegt, z.B.:

Student im Grundstudium =
Student, der im maximal vierten Semester studiert.

$$\mathsf{InBasicStudyPeriod} \equiv \mathsf{Student} \sqcap \forall \mathsf{enrolled\text{-}in}.(\mathsf{Semester} \sqcap \exists \mathsf{has\text{-}number}.\ \leq_4).$$

Werte-Einschränkung von Datentypen

- Im aktuellen Standard: Keine Werte-Einschränkung von (numerischen) Datentypen vorhanden.
- Lösung:
  OWL-Eu (Horrocks u.a.): Erweiterung von OWL um Werte-Einschränkung von numerischen Datentypen
  - Bedeutung für die Modellierung
    - Mehr Möglichkeiten in der Modellierung
    - mehr Einsatzgebiete für OWL
  - Bedeutung für das Schließen
    - Das Schließen bleibt entscheidbar

Arithmetik in OWL

Problem:

Notwendigkeit, arithmetische Zusammenhänge spezifizieren zu können

Beispiel: Ein Student im vierten Semester, der insgesamt weniger als 54 Punkte erbracht hat.

$$\text{Student} \sqcap \exists \text{enrolled-in}.(\text{Semester} \sqcap \exists \text{has-number}.\ =_4 \sqcap \forall \text{has-score}.(\text{BonusAccount} \sqcap \exists \text{crBelow4thSemester}.\ <_{54}))$$

crBelow4thSemester: die Summe aller Punkte, die der Student bis zum vierten Semester erbracht hat

Arithmetik in OWL

- Lösungsvorschlag:
  - Hinzufügen von Arithmetischen Funktionen in OWL
  - Vor dem Schließen: Auswerten der arithmetischen Zusammenhänge

  - Bedeutung für die Modellierung
    - Auch arithmetische Zusammenhänge spezifizierbar
    - Geringere Fehleranfälligkeit durch maschinelle Auswertung arithmetischer Zusammenhänge
  - Bedeutung für das Schließen
    - Das Schließen bleibt unberührt!

„Finite Quantification“

Problem:

Klassendefinitionen mit aufgezählten Klassen können ziemlich ausufern.

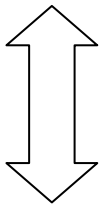
Kleines Beispiel:

$$\begin{aligned}
&\mathsf{StylishWineDrinker} \equiv \mathsf{WineDrinker} \sqcap \\
&\quad [(\exists \mathsf{prefers}.\{\mathsf{red}\} \sqcap \exists \mathsf{prefers}.(\mathsf{Glass} \sqcap \exists \mathsf{suitedFor}.\{\mathsf{red}\})) \sqcup \\
&\quad (\exists \mathsf{prefers}.\{\mathsf{white}\} \sqcap \exists \mathsf{prefers}.(\mathsf{Glass} \sqcap \exists \mathsf{suitedFor}.\{\mathsf{white}\})) \sqcup \\
&\quad (\exists \mathsf{prefers}.\{\mathsf{rose}\} \sqcap \exists \mathsf{prefers}.(\mathsf{Glass} \sqcap \exists \mathsf{suitedFor}.\{\mathsf{rose}\}))] \\
&\mathsf{Wine} \equiv \{\mathsf{red}, \mathsf{white}, \mathsf{rose}\}
\end{aligned}$$

„Finite Quantification“

wünschenswert: Kurznotation für derartige Klassendefinitionen

$$\begin{aligned}
&\text{StylishWineDrinker} \equiv \text{WineDrinker} \sqcap \\
&\quad [(\exists \text{prefers}.\{\text{red}\} \sqcap \exists \text{prefers}.(\text{Glass} \sqcap \exists \text{suitedFor}.\{\text{red}\})) \sqcup \\
&\quad (\exists \text{prefers}.\{\text{white}\} \sqcap \exists \text{prefers}.(\text{Glass} \sqcap \exists \text{suitedFor}.\{\text{white}\})) \sqcup \\
&\quad (\exists \text{prefers}.\{\text{rose}\} \sqcap \exists \text{prefers}.(\text{Glass} \sqcap \exists \text{suitedFor}.\{\text{rose}\}))] \\
&\text{Wine} \equiv \{\text{red}, \text{white}, \text{rose}\}
\end{aligned}$$



$$\text{StylishWineDrinker} \equiv \text{WineDrinker} \sqcap \bigsqcup_{\text{w:Wine}} (\exists \text{prefers}.\{\text{w}\} \sqcap \exists \text{prefers}.(\text{Glass} \sqcap \exists \text{suitedFor}.\{\text{w}\}))$$

„Finite Quantification“

- Lösungsvorschlag:
  - Hinzufügen von einer solchen Kurzschreibweise
  - Vor dem Schließen:
    - Möglichkeit 1: Expansion dieser Kurzschreibweisen in die ursprüngliche OWL-Notation
    - Möglichkeit 2: Erweiterung der OWL-Semantik

- Bedeutung für die Modellierung
  - Kurze und knappe Notation von Klassendefinitionen
  - Geringere Fehleranfälligkeit
  - Bessere Wartbarkeit und Modifizierbarkeit des Modells

„Finite Quantification“

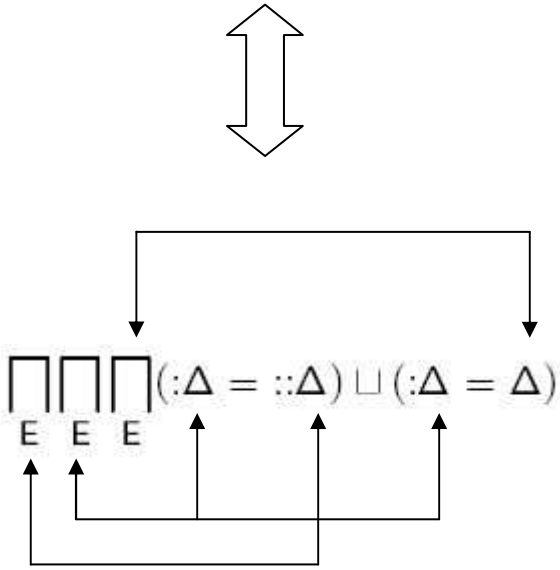
- Bedeutung für das Schließen:
  - Implementierung durch Erweiterung der Semantik von OWL: Optimierung des Schließens
  - Implementierung durch Makros: Keine Änderung des Schlußverfahrens
- Terminierung muss jedoch sichergestellt werden

  → Nur aufgezählte Klassen als Indexklassen erlaubt.
  - keine zyklischen Klassendefinitionen
  - keine (potentiell) unendliche Berechnung der Zugehörigkeit von Individuals zur Indexklasse

„Finite Quantification“

- Variante: Variablenfreie Form der „Finite Quantification“

$$\prod_{a:E}\prod_{b:E}\prod_{c:E}(\{b\} = \{a\}) \sqcup (\{b\} = \{c\})$$



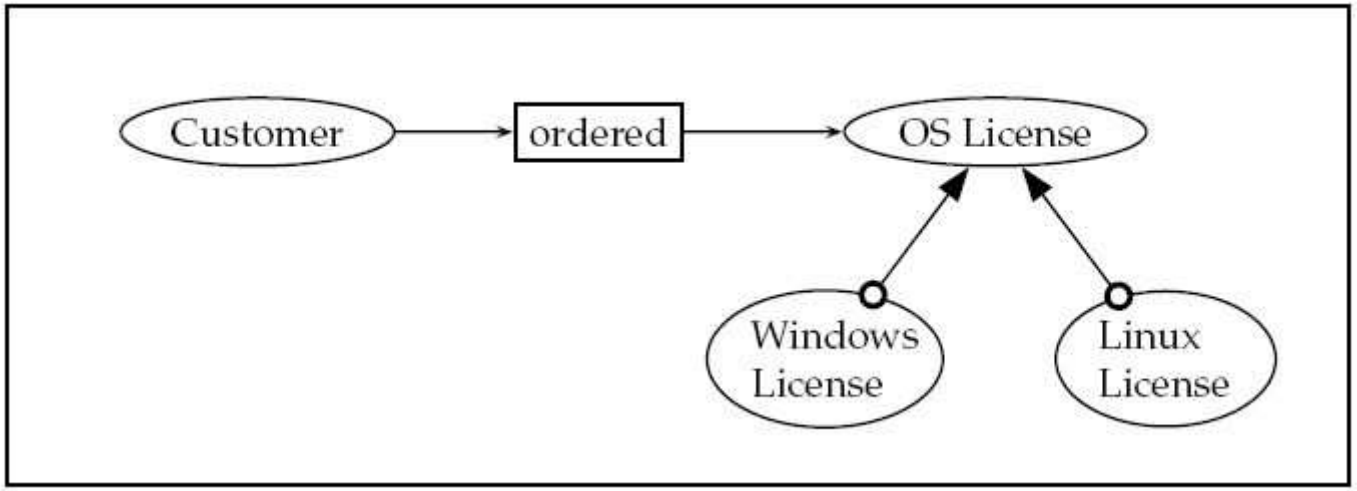
„:“ wählt die „Tiefe“ aus
(von innen aus gesehen)

„Qualified Number Restrictions“

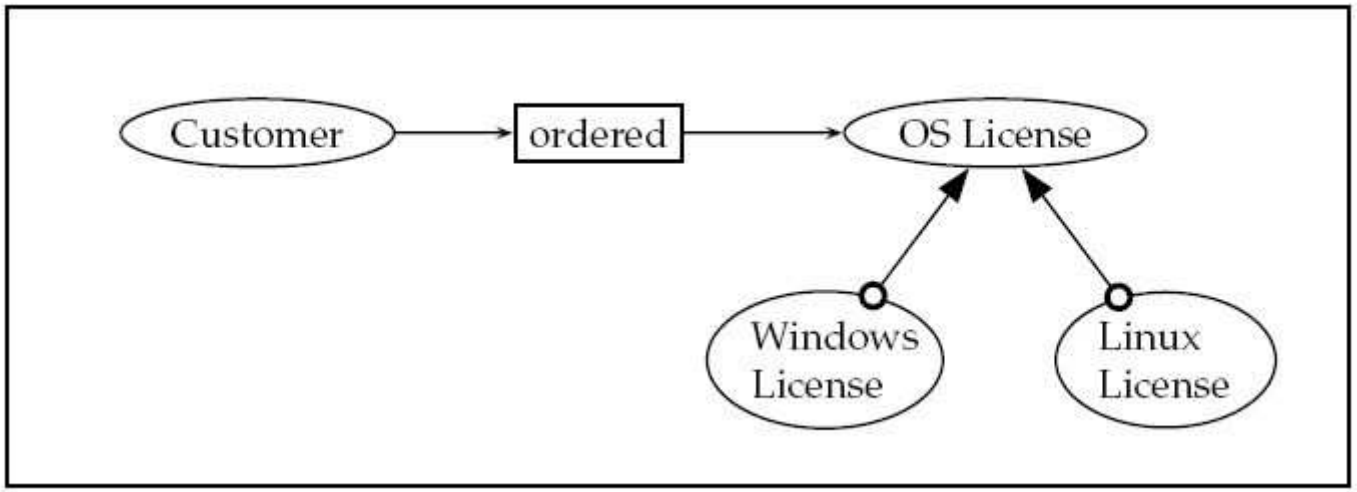
- „Qualified Number Restrictions“:

$$(\leq \text{ n role.C})^{\mathcal{I}} = \{x \mid \#\{y \mid (x, y) \in \text{role}^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \leq n\}$$

- Beispiel:



„Qualified Number Restrictions“

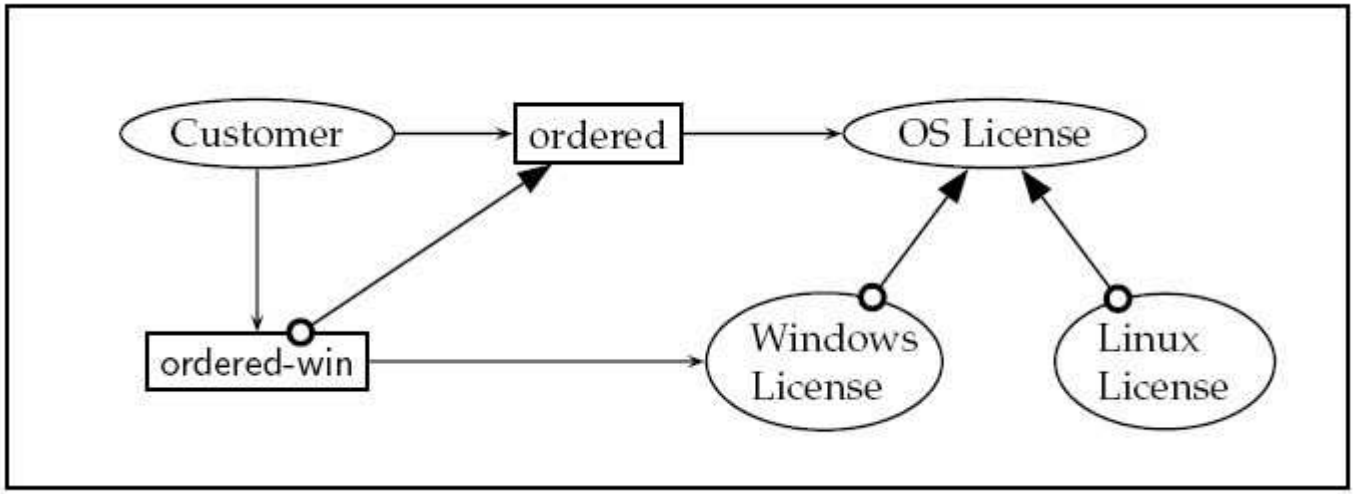


Ein Kunde, der mehr als 20 Windows-Lizenzen bestellt hat (mit „Qualified Number Restrictions“).

$$\text{Windows-Customer} \equiv \text{Customer} \sqcap > 20\,\text{ordered.Windows}$$

„Qualified Number Restrictions“

Notwendige Modifikation des Modells in OWL



Ein Kunde, der mehr als 20 Windows-Lizenzen bestellt hat

(ohne „Qualified Number Restrictions“):

$$\text{Windows-Customer}' \equiv \text{Customer} \sqcap > 20\ \text{ordered-win}$$

„Qualified Number Restrictions“

- Problem:
  - Aufwendige und fehlerhafte Modell-Modifikationen aufgrund fehlender Qualified Number Restrictions
  - Solche work-arounds können zu einem Modell führen, das im allgemeinen nicht korrekt ist.
- Lösungsvorschlag:
  - Hinzufügen der „Qualified Number Restrictions“ in die kommenden Versionen von OWL.

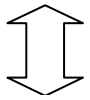
„Qualified Number Restrictions“

- Bedeutung für die Modellierung:
  - Keine aufwendigen und fehleranfälligen work-arounds mehr
  - Keine proprietären OWL-Varianten, wie es aktuell der Fall ist
- Bedeutung für das Schließen:
  - Theoretisch keine Änderung der Komplexität durch „Qualified Number Restrictions“
  - Jedoch Szenarien möglich, in denen die Performanz sehr leidet
  - Aber: Schließen mit „Qualified Number Restrictions“ kann optimiert werden

Zur Verwendung von SWRL

- SWRL: Semantic Web Rule Language
  - Sprache, die OWL um Horn-artige Regeln erweitert
  - Ähnlich zu Sichten in Datenbanksystemen
  - Gestattet die Verwendung von beliebig vielen Variablen (im Gegensatz zu OWL: Nur zwei Variablen in einem Gültigkeitsbereich)
  - Unentscheidbare Teilmenge der Prädikatenlogik
- Beispiele

$$\text{TakenExam} \equiv \text{Exam} \sqcap \exists \text{has-result}.\text{ExamResult}$$



$$\text{Exam}(?x) \wedge \text{has-result}(?x, ?y) \wedge \text{ExamResult}(?y) \rightarrow \text{TakenExam}(?x)$$

Zur Verwendung von SWRL

→ SWRL ist ausdrucksmächtiger

Folgende SWRL-Regeln besitzen kein OWL-Gegenstück

$$\text{father-of}(?dad, ?kid) \wedge \text{brother-of}(?dad, ?uncle) \rightarrow \text{uncle-of}(?uncle, ?kid)$$

$$\text{precedes}(?s1, ?s2) \wedge \text{precedes}(?s2, ?s3) \rightarrow \text{precedes}(?s1, ?s3)$$

$$\text{Sender}(?s) \wedge \text{Receiver}(?r) \wedge \text{Channel}(?c) \wedge \text{endpoint}(?c, ?s) \wedge \text{endpoint}(?c, ?r) \wedge \text{frequency}(?s, ?f1) \wedge \text{frequency}(?r, ?f2) \rightarrow \text{builtIn}(\text{equals}, ?f1, ?f2)$$

Zur Verwendung von SWRL

- Probleme bei der Modellierung mit SWRL
  - Übermäßiger Gebrauch von Variablen macht Auffinden möglicher OWL-Gegenstücke zu den Regeln schwer
  - Regelsprachen als intuitivere Formalismen evtl. durch Modellierer bevorzugt
  - Sorgfalt bei der Verwendung von SWRL notwendig, um Entscheidbarkeit zu erhalten
  - Modellierer muss sich beider Modellierungssprachen bewusst sein.
- Probleme beim Schließen mit SWRL
  - SWRL ist prinzipiell unentscheidbar
  - Zusammenarbeit von OWL-Reasoner und SWRL-Reasoner nicht (ausreichend) geklärt

Zur Verwendung von SWRL

- Arten der Zusammenarbeit von OWL- und SWRL-Reasoner
  - Gar keine: Beide Reasoner bearbeiten die Wissensbasis isoliert voneinander
  - Beide Reasoner machen ihre Ergebnisse dem anderen zugänglich
    - Schließen nach dem Sättigungsverfahren
    - SWRL-Reasoner kann Ergebnisse des OWL-Reasoners problemlos übernehmen
    - Ergebnisse des SWRL-Reasoners müssen zu den Instanzen und Hierarchiebeziehungen in OWL hinzugefügt werden
  - Konflikterkennung und -auflösung notwendig
  - Terminierung des Sättigungsverfahrens muss garantiert werden
- Trennung von OWL und SWRL, um Interoperabilität zu erhöhen

Zusammenfassung

- OWA, CWA für die Modellierung in OWL sinnvoll
- Werte-Einschränkung numerischer Datentypen erhöht die Flexibilität von OWL und erhält die Entscheidbarkeit
- Hinzufügen von Arithmetischen Funktionen und deren Auswertung vor dem Schließen erlaubt eine vollständigere Modellierung und beeinträchtigt das Schließen nicht
- „Finite Quantification“ erleichtert Notationen von Klassendefinitionen durch aufgezählte Klassen und erhöht Lesbarkeit, Wartbarkeit und Modifizierbarkeit der OWL-Wissensbasis

Zusammenfassung

- „Qualified Number Restrictions“ vermeiden fehleranfällige und inkorrekte Modelle, wobei das Schließen durch Optimierungen weiterhin performant bleibt
- SWRL ist eine Erweiterung von OWL mit Regeln. Die Kooperation von SWRL- und OWL-Reasoner kann auf unterschiedliche Weise gestaltet werden: isoliert oder zusammen mit dem Sättigungsverfahren