

# Znalostní vzory – Knowledge patterns

Rostislav Miarka

Katedra informatiky a počítačů, Přírodovědecká fakulta,  
Ostravská Univerzita v Ostravě, 30. dubna 22, 701 03, Ostrava  
Rostislav.Miarka@osu.cz

**Abstrakt.** Příspěvek popisuje techniku zvanou znalostní vzory, která se používá při budování ontologií. Pomocí znalostních vzorů se vytvářejí obecná schémata znalostí, která se následně transformují do slovníku určité specifické domény. Podobným přístupem jako znalostní vzory je přístup dědičnosti, který má ovšem některá omezení. Použití znalostních vzorů usnadňuje explicitní vyjádření obecných teorií a pomocí transformací umožňuje jejich opětovné použití při budování různých znalostníchází. Součástí příspěvku jsou také příklady, které pomáhají pochopit popisovanou problematiku.

**Klíčová slova:** znalostní vzory, dědičnost, budování ontologií

## 1 Úvod

Znalostní vzory je možné použít při budování formální ontologie. Jsou založeny na rozpoznání a explicitním vyjádření opakujících se vzorů znalostí v ontologii a jejich následném namapování na specifické koncepty v ontologii. Znalostní vzory vytvářejí jak abstraktní teorie, tak jejich mapování do určité specifické domény. Z výpočetního hlediska poskytují znalostní vzory efektivní mechanismus pro opětovné použití znalostí.

Při budování formální ontologie se znalostní inženýr (případně jiná osoba budující ontologii) pokouší identifikovat koncepty a axiomy popisující svět, o který se zajímá. Běžně se stává, že při této činnosti nalézá vzory axiómů s podobnou strukturou. Znalostní vzory se používají k explicitnímu vyjádření těchto obecných šablon označujících opakující se schémata v ontologii. Součástí znalostních vzorů jsou také jejich transformace, pomocí kterých se vytváří specifické teorie. Tyto transformace probíhají skrze přejmenování symbolů. [1]

## 2 Dědičnost a její omezení

Při vytváření specifické teorie z obecné teorie se nabízí použití mechanismů dědičnosti (např. pomocí relace ISA). Dědičnost má ovšem některá omezení. Prvním omezením je, že pokud axióm z obecné teorie platí pro více různých konceptů ze specifické teorie, může se stát, že dojde k promíchání jednotlivých konceptů v axiómech a v důsledku i k vyvozování nesmyslných závěrů.

Například v ontologii týkající se zaměstnanců výrobního podniku se budou vyskytovat axiomy týkající se toho, jak se stanoví mzda jednotlivých zaměstnanců. Dělníci mohou mít složenou svoji mzdu z hodinové a úkolové mzdy. Jedním z axiómů je, že měsíční hodinová mzda je součinem hodinové sazby a počtu odpracovaných hodin za měsíc. Dalším axiómem je, že měsíční úkolová mzda je součinem sazby za vyrobené kusy a počtu vyrobených kusů za měsíc. Všechny uváděné axiomy budou zapsány v jazyce PROLOG.

```
% výpočet měsíční hodinové mzdy určité osoby (dělníka)
hodinova_mzda(Osoba, H) :-
    isa(Osoba, delnik),
    hodinova_sazba(Osoba, HS),
    odpracovane_hodiny(Osoba, OH),
    H is HS * OH.

% výpočet měsíční úkolové mzdy určité osoby (dělníka)
ukolova_mzda(Osoba, U) :-
    isa(Osoba, delnik),
    sazba_za_ks(Osoba, SK),
    vyrobené_ks(Osoba, VK),
    U is SK * VK.
```

Obecně tedy pro oba druhy mzdy platí, že se vypočte jako součin sazby za jednotku a počtu jednotek.

```
mesicni_mzda(X, Y) :-
    isa(X, CLASS),
    sazba_za_jednotku(X, S),
    pocet_jednotek(X, P),
    Y is S * P.
```

Po zavedení obecného axiómu do uvedené ontologie tedy vznikne axióm:

```
% výpočet měsíční mzdy zaměstnance
mesicni_mzda(Zamestnanec, MM) :-
    isa(Zamestnanec, zamestnanec),
    sazba_za_jednotku(Zamestnanec, SJ),
    pocet_jednotek(Zamestnanec, PJ),
    MM is SJ * PJ.
```

V ontologii nejsou axiomy vyjadřující vztahy mezi objekty – např. počet odpracovaných hodin není počet vyrobených kusů. Při použití dědičnosti tedy může být použita hodinová sazba k výpočtu úkolové mzdy a naopak.

Tomuto problému se lze vyhnout několika způsoby. Může se stanovit, že axiomy z obecné teorie budou použity ve specifické teorii pouze jedenkrát. Nebo se může do obecných axiómů přidat parametr, který ve specifické teorii určuje, o jaký koncept se jedná. V tomto případě se ale může stát, že se musí přidávat stále nové parametry, aby bylo možné rozlišit koncepty, kterých se axióm týká. Když se řekne, že koncept ze specifické teorie *je (is a)* konceptem z obecné teorie, je tím myšleno, že první koncept

*může být formulován jako* druhý koncept. Dědičnost ovšem provádí přenos axiómu pomocí vztahu *isa*, ne pomocí vztahu *může být formulován jako*. [1] V praxi ovšem často dochází ke sjednocení těchto dvou vztahů, protože dědičnost je často jediný způsob, jak se přiblížit k reprezentaci specifické teorie pomocí obecné teorie.

### 3 Znalostní vzory jako teorie prvního řádu

Při budování znalostníchází pomocí znalostních vzorů se tedy definuje vzor jako teorie prvního řádu, jejíž axiomy nejsou součástí cílové znalostní báze, ale mohou být do ní začleněny pomocí přejmenování jejich nelogických symbolů. Vzor se implementuje jako explicitní a samostatná teorie. Potom se stanoví transformace (pravidla přejmenování symbolů) pro každou zamýšlenou aplikaci tohoto vzoru do cílové znalostní báze. Transformace je konzistentní mapování nelogických symbolů ze vzoru na termy znalostní báze, které říká, jak má být vzor transformován.

V případě již zmíněné ontologie se tedy použije obecný axiom pro výpočet měsíční mzdy libovolného zaměstnance, který je uveden v předchozí kapitole. Pro tento axiom se zavedou transformační pravidla, která ho namapují na objekty specifické domény. První pravidlo bude:

```
zamestnanec -> delnik
mesicni_mzda -> hodinova_mzda
sazba_za_jednotku -> hodinova_sazba
pocet_jednotek -> odpracovane_hodiny
isa -> isa
```

a druhé bude:

```
zamestnanec -> delnik
mesicni_mzda -> ukolova_mzda
sazba_za_jednotku -> sazba_za_ks
pocet_jednotek -> vyrobene_ks
isa -> isa
```

Poslední řádek (mapování symbolu na sebe samého) je zde proto, že ve vzoru mohou existovat symboly, které nelze namapovat na objekty ze specifické domény. Pokud nastane tento případ, vynechají se při transformaci tyto symboly a každý z nich se namapuje na nový jedinečný symbol. Symboly jsou po provedení transformace součástí znalostní báze, ale jsou skryté ostatním axiomům ve znalostní bázi. [1]

#### 3.1 Možné postupy při použití znalostních vzorů

Prvním možným postupem při použití znalostních vzorů je, že znalostní vzor je do znalostní báze zahrnut pomocí syntaktického procesu přejmenování symbolů – již uvedené transformace. Při tomto procesu se převádí prostřednictvím přejmenování

symbolů slovník (ontologie) použitý ve vzoru do aplikační domény, ve které má být vzor použit.

Dalším možným postupem je řešení specifického problému pomocí ekvivalentního (izomorfního) problému. V tomto případě se slovník doménově specifického problému převede tak, aby byl použitelný ve vzoru (vytvoří se zobecněný problém řešitelný pomocí vzoru), vyřeší se pomocí vzoru a výsledek se převede zpět do slovníku specifické domény.

Pokud je znalostní vzor spolu s mapovacími axiomy přidán přímo do znalostní báze, výsledek bude stejný, jako kdyby zde byla přidána již transformovaná verze tohoto vzoru. Nevýhodou je, že mapování je implementačně složitější a více výpočetně náročnější. Výhodou ovšem je, že znalostní vzor je přímo explicitně vyjádřen ve znalostní bázi, není zde jen jeho transformovaná kopie.

#### 4 Výhody a nevýhody znalostních vzorů

Hlavní výhodou použití znalostních vzorů je, že jejich princip je velmi jednoduchý – vytvoření obecných vzorů znalostí a jejich následné namapování na specifickou doménu.

Nevýhodou je, že znalostní vzory neřeší mapování obecných axiomů na axiomy specifické domény. Vhodné mapování příslušných nelogických symbolů musí určit znalostní inženýr.

#### Reference

1. In Staab, S., Studer, R. (eds). *Handbook on Ontologies*. Springer, 2004, Springer-Verlag Berlin. ISBN 3-540-40834-7.

#### Annotation:

##### *Knowledge patterns*

This paper describes a technique called knowledge patterns, which is used at building ontologies. By the help of knowledge patterns they create general schemata of knowledge, which are then transformed into dictionary of specific domain. The similar approach to knowledge patterns is an approach of inheritance which has indeed some limitations. Using knowledge patterns simplify explicit representation of general theories and by the help of transformations enable their reuse at building various knowledge bases. Parts of the paper are also examples to help understand described problems.