Obsah

1 Úvod

2 Teorie k řešené problematice

2.1 Reprezentace znalostí

2.2 Popis obrázků

2.3 Sémantika

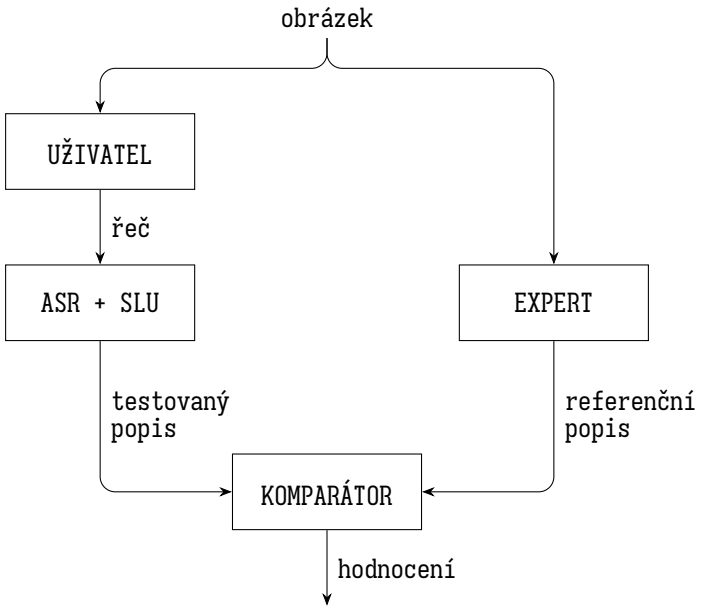
3 Návrh a architektura systému

3.1 Obecná architektura systému

Obecná architektura celého systému vychází z jeho požadované funkčnosti, kterou je porovnání obrázku s jeho popisem v přirozené řeči, a to na sémantické úrovni. Z toho pak plyne, že celý systém se ve své podstatě skládá ze tří základních částí:

1. referenční (vzorový) popis daného obrázku
2. sub-systém pro zpracování přirozené řeči (⇒ testovaný popis)
3. porovnání referenčního a testovaného popisu

Jednotlivé části spolu vzájemně fungují následujícím způsobem: Uživateli je prezentován obrázek a jeho úkolem je popsat, co na obrázku vidí. Získaný popis v přirozené řeči je převeden na text (ASR). [TODO1] Z tohoto přepisu je extrahována sémantická informace (SLU), ze které je vytvořen testovaný popis. Testovaný popis je porovnaný s referenčním (vzorovým) popisem daného obrázku. Výsledek tohoto porovnání lze pak považovat za finální výstup, ale také je možné jej použít jako vstup pro další zpracování (např. vektor příznaků pro klasifikátor). Schématické znázornění je na Obrázku 1.



Obrázek 1: Schéma obecné architektury systému

[1] Jak/kde vysvětlit zkratky?

Pro tuto práci bylo rozhodnuto, že základem bude expertní přístup. Od toho se poté odvíjí konkrétní algoritmy, formáty a postupy navržené a implementované v této práci, které jsou podrobněji popsány v pozdějších kapitolách. Je ale vhodné zmínit, že během návrhu bylo dbáno na to, aby bylo možné pro reálná nasazení některé implementace v případě potřeby zaměnit nebo upravit, aby lépe vyhovovaly specifickým požadavkům pro dané použití.

TODO2

3.2 Reprezentace znalostí

Pro porovnání obrázku s jeho popisem v přirozené řeči bylo nutné zvolit či navrhnout nějakou formu reprezentace znalostí, která by umožnila zachytit sémantiku z obou zdrojů. Jak již bylo výše zmíněno, zvolen byl expertní přístup a to v tomto případě znamená, že referenční popis obrázku je vytvořen lidským expertem, což klade další omezení na formát reprezentace znalostí.

Při návrhu bylo tedy potřeba brát v úvahu následující požadavky a najít nějaký formát, který by představoval vhodný kompromis mezi nimi.

- **Čitelnost člověkem**:
  Aby byl lidský expert schopen vytvořit, číst a případně upravit referenční popisy, je nutné, aby byl schopen porozumět formě a zápisu uložených dat. Toto omezení tedy upřednostňuje textové formáty a prakticky vyřazuje binární data.

  Výjimku by mohl tvořit nějaký binární formát s přidruženým editorem, kde by člověk mohl v grafickém prostředí prohlížet a manipulovat data, ale takový případ je nad rámec této práce.

- **Kompaktnost a struktura dat**:
  Dalším důležitým aspektem je struktura a kompaktnost dat. Pomocí počítače je poměrně snadné v krátkém čase zpracovat velké množství jednoduchých datových záznamů, nicméně člověk se bude lépe orientovat v nějakém kompaktnějším popisu, který ačkoli může být složitější ve své struktuře, tak bude pro člověka lépe názorný a uchopitelný.

[2] zmínit, že referenční popis není třeba pokaždé tvořit znovu, ale lze udělat „offline“ předem?

- **Univerzálnost formátu**:
  Podstatnou vlastností pro hledanou reprezentaci znalostí je její schopnost zachytit popis různých obrázků. Navržený formát tedy musí být dostatečně univerzální, aby pomocí něj šlo popsat co nejširší spektrum informací, od jednoduchých obrázků zobrazujících například jeden statický objekt, přes složitější obrázky zobrazující více objektů, až po dynamické komplexní scény zobrazující mnoho objektů, činnosti a vazby mezi nimi.

- **Počítačová zpracovatelnost**:
  V neposlední řadě je také potřeba dbát na to, aby navržený formát bylo možné co nejsnadněji zpracovat programově, na počítači. Dynamické formáty s volnou strukturou bývají složitější na strojové zpracování, než fixní formáty s přesně definovanou podobou.

TODO3

S ohledem na tyto body byla navržena reprezentace znalostí založená na sémantických sítích, která definuje 4 základní aspekty popisu: objekty, jejich hierarchii, statické atributy a dynamické vazby. Detailnější popis těchto jednotlivých aspektů je v následujících částech, konkrétní technická implementace je pak popsána v sekci 4.

TODO4

TODO5

3.2.1 Objekty ve scéně

Cokoli, co lze v obrázku ohraničit rámečkem (angl. bounding-box) a při separaci od zbytku scény (obrázku) neztratí nebo zásadně nezmění svůj význam, lze považovat za *objekt*.

Jako *objekt* v obrázku lze tedy označit zobrazené fyzické předměty, postavy, zvířata, ale také nehmotné pojmy jako „nebe“, místa, lokace či místnosti (např. „kuchyň“ nebo „louka“) a části jiných objektů (např. „obličej“ jsou součást hlavy nebo celého člověka).

[3] zmínit, že expertní přístup umožňuje lepší kontrolu nad obsahem/kvalitou než automat/statistika?

[4] jak s anglikanismy?

[5] jak a kde použít emph?

Tato definice objekt u byla záměrně navržena velmi obecně, aby byl definovaný formát univerzální a šel použít i pro popis velmi odlišných obrázků s různými účely. Potenciální nevýhodou, která plyne z univerzálnosti formátu, může být v některých situacích problém nejednoznačnosti.

V jednom obrázku lze definovat různé množiny objektů, podle toho, jak moc detailní popis expert vytvoří. Například pokud by byl na obrázku člověk, lze jej popsat jedním objektem jako „člověk“ nebo „osoba“, ale také by šlo definovat ještě mnoho dalších objektů, například pro jednotlivé části těla nebo oblečení.

Kromě různých úrovní detailů lze také na problematiku nejednoznačnosti narazit v situaci, kdy je pro nějaký (dostatečně komplexní) obrázek možné sestavit různé množiny objektů podle toho, pro jaké potřeby je zrovna obrázek a referenční popis používán. Pro aplikaci, kde je podstatné zachycení živých objektů, může být množina objektů v referenčním popise tvořena lidmi či zvířaty. Pro jiné použití pak ale může být podstatné zachytit prostředí a neživé předměty, takže množina objektů by byla tvořena částmi prostředí (např. stromy, křoví, voda, skály), budovami nebo obecnými předměty.

TODO6

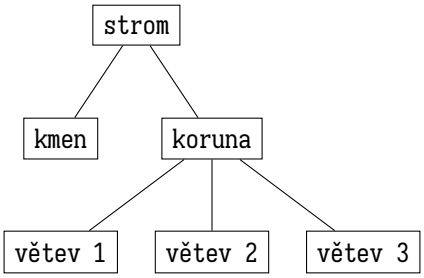
3.2.2 Hierarchie objektů

Kromě množiny samotných objektů lze v obrázku také definovat jejich hierarchii. To přirozeně plyne z výše zmíněné definice *objektu*, která umožňuje specifikovat část existujícího objektu jako další samostatné objekty.

Příkladem takového popisu může být například situace, kdy je na obrázku strom. Strom je možné rozdělit na korunu a kmen, korunu pak je možné dále dělit na větve. Schématicky lze tento popis znázornit jako stromovou strukturu, viz Obrázek 2.

Z pohledu daného objektu jsou „vyšší“ (obecnější) objekty označované jako *rodičovské objekty* (nebo jen *rodiče*) a „nižší“ (konkrétnější) objekty pak jako *potomci*. Aby byl popis jednoznačný, tak bylo rozhodnuto, že vazby musejí být definované v referenčním popisu oboustranně. Tím je myšleno to, že pokud objekt `A` specifikuje objekt `B` jako svého rodiče, pak musí i objekt `B` specifikovat `A` jako svého potomka. Pokud je hierarchická vazba

[6] vyměnit dlouhé popisy za jeden ukázkový příklad?



Obrázek 2: Schématické znázornění hierarchie objektů

definována pouze jednostranně, měl by být referenční popis implementací systému odmítnut jako invalidní.

Možnost definovat hierarchii objektů nabízí mimo jiné také způsob, jak vytvořit skupiny objektů, které k sobě nějakým způsobem patří. Například strom může být součástí lesa, kráva může být součástí stáda nebo postava na hřišti může být součástí fotbalového týmu. Tato příslušnost objektu nějaké skupině je dalším typem sémantické informace, kterou umožňuje navržený formát zachytit bez nutnosti definovat další specializované struktury.

Mohou však nastat situace, kdy je potřeba jeden objekt zařadit do několika různých skupin. Z tohoto důvodu bylo rozhodnuto, že každý objekt může mít libovolné množství rodičů a libovolné množství potomků. TODO7

3.2.3 Atributy objektů

Vedle pouhého výčtu samotných objektů ve scéně je dále přirozeným požadavkem, aby byla reprezentace znalostí schopna zachytit i jejich vlastnosti. K tomu slouží třetí aspekt popisu - *atribut*.

Atributem je možné popsat jakoukoli informaci o objektu, která není závislá na jiném objektu. Jinými slovy, pokud bychom objekt izolovali od zbytku scény (obrázku), tak všechny vlastnosti, které se tím nezmění nebo nezaniknou, lze popsat pomocí atributů. Typickým příkladem je barva nebo tvar objektu či jeho části.

Atribut se skládá ze zvoleného *názvu* a přiřazené *hodnoty*, kdy konkrétní názvy a hodnoty jsou volbou experta, který tvoří referenční popis. Název atributu označuje jakou

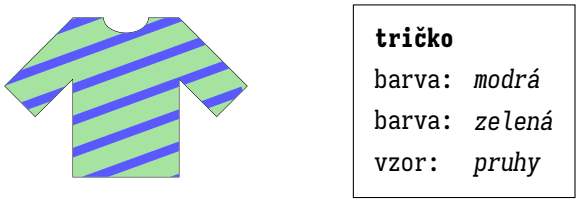
[7] zdůvodňovat? dávat příklad? přirovnávat k OOP dědičnost vs kompozice?

vlastnost daný atribut popisuje a přiřazená hodnota pak udává, jaké konkrétní hodnoty nabývá. Každý objekt může mít libovolné množství těchto atributů.

Názvy atributů pod jedním objektem nemusí být unikátní, lze specifikovat víc stejnojmenných atributů s různými hodnotami. Typickým příkladem může být vícebarevný objekt, kde atribut s názvem „barva“ bude vícekrát, pro různé konkrétní barvy. Příklad jednoho takového popisu je na Obrázku 3, kde název objektu je „tričko“ a definované jsou tři atributy: dvě barvy a vzor.

TODO8

TODO9

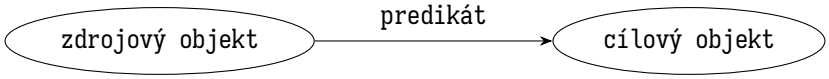


Obrázek 3: Příklad použití atributů pro popis objektu

3.2.4 Vazby mezi objekty

Posledním typem informace, kterou by měl být referenční popis obrázku schopen zachytit, jsou vazby a vztahy mezi různými objekty. Může se jednat například o činnosti, které se týkají dvou objektů, nebo popis relativních vlastností, jako je velikost či pozice.

Každý takový záznam je označen jako *triplet* a skládá se ze *zdrojového objektu*, *cílového objektu* a *predikátu*, který popisuje danou vazbu nebo vztah. Tato struktura je inspirována běžným zápisem sémantických sítí a RDF standardem. TODO10 Obecné schéma jednoho tripletu je na Obrázku 4.



Obrázek 4: Obecné schéma tripletu

[8] hezčí tabulka pro popis objektu?

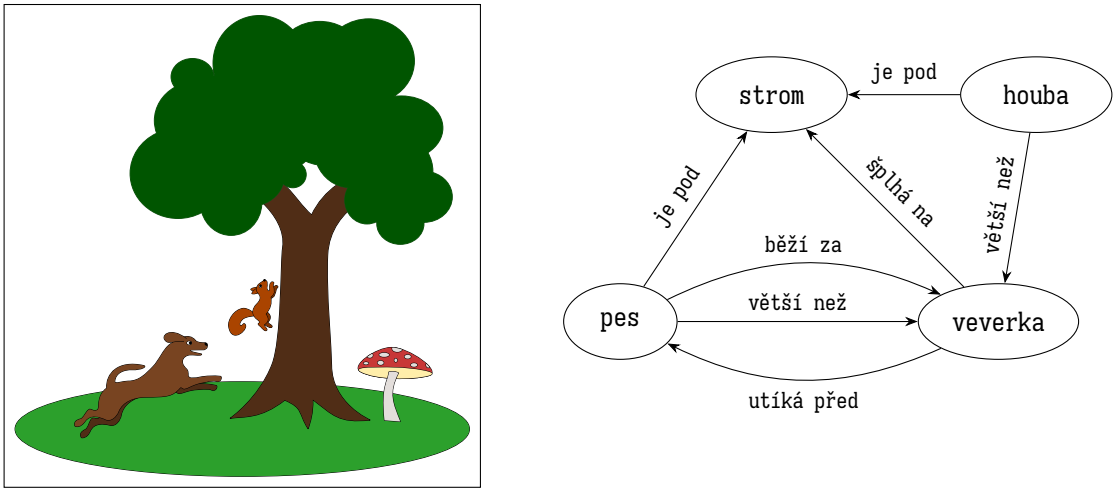
[9] přidat šipky a popisky do tabulky co je co?

[10] doplnit reference!

Každý objekt může být součástí libovolného počtu tripletů a to jak v pozici zdrojového, tak cílového objektu. Objekt také nemusí být součástí žádných vazeb a dokonce v celém popisu nemusí být žádné vazby definované. Například pokud bychom chtěli pouze testovat paměť uživatele, mohli bychom mu ukázat obrázek, pak jej skrýt a sledovat, jaké objekty si vybaví. V takové úloze nás vazby mezi objekty vůbec nemusejí zajímat a je tedy zbytečné, aby byly součástí referenčního popisu.

Při vytváření referenčního popisu obrázku se opět naskýtá otázka, jaké vazby a vztahy mezi objekty do popisu zavést a které lze ignorovat. Například relativní velikost nebo pozice objektů může být pro některé úlohy klíčová, ale pro jiné zcela nepodstatná. Tato problematika byla opět ponechána na expertovi, který vytváří referenční popis pro danou konkrétní aplikaci, aby rozhodl, které informace je potřeba v referenčním popise zachytit.

Konkrétní příklad několika tripletů je znázorněn na Obrázku 5.



Obrázek 5: Příklad použití tripletů pro popis vztahů mezi objekty

TODO11

3.3 Extrakce sémantické informace

Dalším klíčovým bodem při návrhu systému pro hodnocení popisu obrázku bylo najít způsob, jak z přirozené řeči extrahovat požadovanou sémantickou informaci. Cílem je

[11] příklad s kompletním popisem obrázku (všechno dohromady)?

z přirozené lidské řeči získat informace v takové podobě, aby je bylo možné porovnat s naším referenčním popisem obrázku a následně vyhodnotit jejich podobnost.

TODO12 Extrakce sémantiky z přirozené řeči se běžně dělá z textového přepisu dané promluvy. I v této práci tedy extrakce sémantických informací probíhá z textu. To znamená, že pokud uživatel popíše obrázek mluvenou řečí, tak je potřeba promluvu převést do textu pomocí nějakého systému pro rozpoznání řeči (ASR). TODO13 TODO14 Problematika rozpoznání řeči a převodu audia do textu je nad rámec této práce a předpokládá se, že bude v praxi řešena nějakou již existující implementací. Ve zbytku práce bude tedy pro zjednodušení rovnou předpokládaným vstupem text. TODO15

Otázka extrakce sémantické informace se tedy zužuje na extrakci sémantiky z přirozené řeči v textové podobě. Jako způsob řešení byl zvolen přístup založený na sémantickém parsování pomocí bezkontextových gramatik, který je v souladu s volnou expertního přístupu v celé této práci.

Základní koncept celého sub-systému pro extrakci sémantiky byl navržen tak, že podle referenčního popisu obrázku bude expertem sestavena gramatika, podle které budou v textu detekované jednotlivé sémantické entity. Gramatika je v tomto kontextu sada pravidel, která definují, jaké promluvy jsou v textu očekávané. Dále tato pravidla také udávají informace o tom, jakou sémantickou entitu daná detekovaná promluva vyjadřuje. Konkrétní syntaxe, použití a funkčnost těchto gramatik bude popsán později spolu s implementací v části 4.1.

Tyto sémantické entity pak budou porovnané s referenčním popisem

3.3.1 Sémantické entity

První otázkou, kterou bylo potřeba vyřešit pro získání sémantické informace z přirozeného popisu, byla její podoba. Jinými slovy, jak by měla extrahovaná sémantika vypadat, aby ji bylo možné porovnat s referenčním popisem obrázku.

[12] ozdrojovat?

[13] existují metody co to dělají rovnou z audia?

[14] změna času? (z minulého do přítomného)

[15] doplnit, že jsem při vývoji používal ruční přepisy? Případně zmínit odkud jsem je vzal?

Vzhledem k tomu, že výše definovaný referenční popis (viz sekce 3.2) se skládá z objektů, jejich hierarchie, atributů a vazeb, tak se nabízí přímo tyto čtyři typy informací hledat v textu.

Bylo tedy rozhodnuto, že z přepisu přirozené řeči budou extrahované:

- objekty →systém pro extrakci sémantiky byl měl být schopen v textu detekovat všechny zmíněné objekty, které jsou zároveň definované v referenčním popisu.
- atributy/vlastnosti objektů
- vazby/vztahy mezi objekty

3.4 Hodnotící algoritmus

4 Implementace a testování

4.1 Sémantické parsování pomocí gramatik

5 Závěr