

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Problematika reprezentace znalostí a popisu obrázku</b>	<b>3</b>
2.1	Reprezentace znalostí . . . . .	3
2.2	Popis obrázků . . . . .	3
2.3	Sémantika . . . . .	3
<b>3</b>	<b>Návrh a architektura systému</b>	<b>4</b>
3.1	Obecná architektura systému . . . . .	4
3.2	Reprezentace znalostí . . . . .	5
3.3	Extrakce sémantické informace . . . . .	7
3.4	Hodnotící algoritmus . . . . .	7
<b>4</b>	<b>Implementace a testování</b>	<b>8</b>
<b>5</b>	<b>Závěr</b>	<b>8</b>

# 1 Úvod

## **2 Problematika reprezentace znalostí a popisu obrázku**

### **2.1 Reprezentace znalostí**

### **2.2 Popis obrázků**

### **2.3 Sémantika**

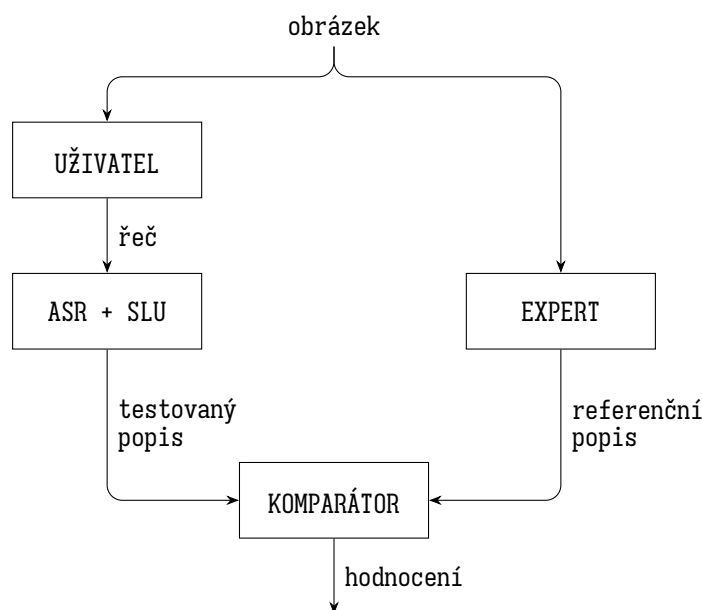
## 3 Návrh a architektura systému

### 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 ( $\Rightarrow$  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). Z tohoto přepisu je extrahována sémantická informace (SLU), ze které je vytvořen testovaný popis. Testovaný popis je porovnán 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

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.

## 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.

- **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.

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, spolu s ukázkami na konkrétních příkladech a obrázcích.

## 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 součástí hlavy nebo celého člověka).

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

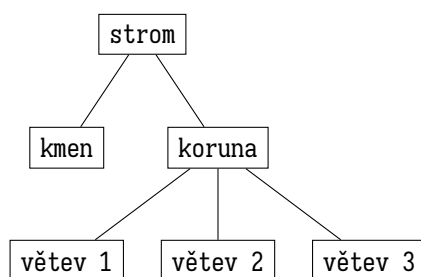
V jednom obrázku lze definovat různé množiny objektů, podle toho, jak moc detailní popis expert vytvoří. Například pokud by byl na obrázku člověk, lze jej popsat jedním objektem jako „člověk“ nebo „osoba“, ale také by šlo definovat ještě mnoho dalších objektů, například pro jednotlivé části těla nebo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

## 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.



Obrázek 2: Schématické znázornění hierarchie objektů

Z pohledu daného objektu jsou „vyšší“ (obecnější) objekty označovány 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 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.

#### **Statické atributy objektů**

#### **Dynamické vazby mezi objekty**

### **3.3 Extrakce sémantické informace**

### **3.4 Hodnotící algoritmus**

## **4 Implementace a testování**

## **5 Závěr**