

## mm-cat

---

<https://www.ksi.mff.cuni.cz/~koupil/mm-cat/index.html>

ER schéma <- převede do -> unified categorical representation (jako SK v evocat, update asi ne jako SK evocat)

Podpora 2 DBMS (Schema-less (MongoDB) vs. schema-full/schema-mixed (PostgreSQL)).

potom lze z UCR vytvořit pro PostgreSQL instance category (a můžu nad ní dělat dotazy SELECT, ostatní ne, ale je plánováno rozšíření)

nebo lze z UCR vytvořit pro MongoDB migraci dat do MongoDB

## mm-infer

---

framework, co umí odvození společného schématu multi-model dat (framework that ensures **inference** of a common schema of multi-model data)

nástroj pro odvozování struktury dat (tj. ne všechna data mají předem definované schéma a pomocí tohoto nástroje dokážeme schéma zpětně rekonstruovat už z uložených dat - používá se hlavně v dokumentových a grafových DBMS)

mám různé databáze - zadám do mm-infer, ten to zchroustá a vyplivne návrh v koncepčním schématu (schema + node part -> v prezentaci), pak můžu exportovat ze schéma do nějakého formátu (JSON, xml a tak)

dokáže zpracovat hodně dat

## mm-quecat

---

<https://github.com/yawnston/queycat>

umí: query-data (nezávisle na databázi pod tím)

category theory: sjednocená abstraktní reprezentace -> můžu se na to dívat jako na graf -> používat SPARQL-based query language

importovaná reprezentace SK od evocat

aplikovaný jazyk podobný SPARQL: **MMQL**

# evocat

<https://nosql.ms.mff.cuni.cz/mmcat/> - samotná aplikace

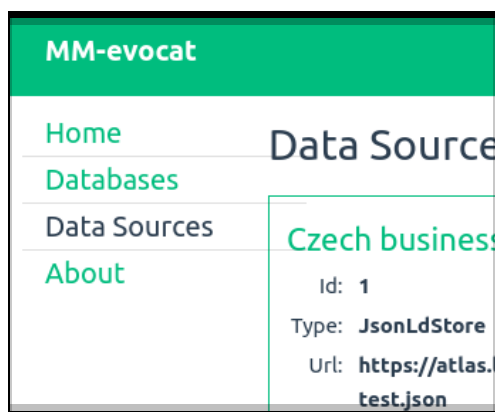
<https://mm-evocat.com/> - rozcestník (i na dokumentaci)

<https://gitlab.mff.cuni.cz/contosp/evolution-management> - Gitlab

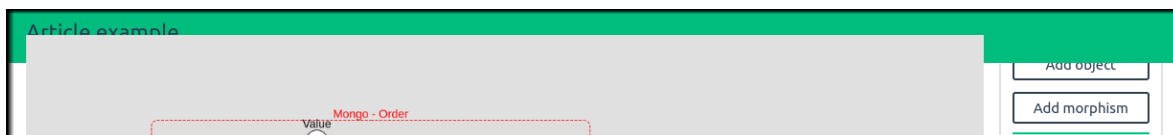
## Uživatelské hledisko - obal

(běžné, co dělá aplikace, seznamování se s aplikací)

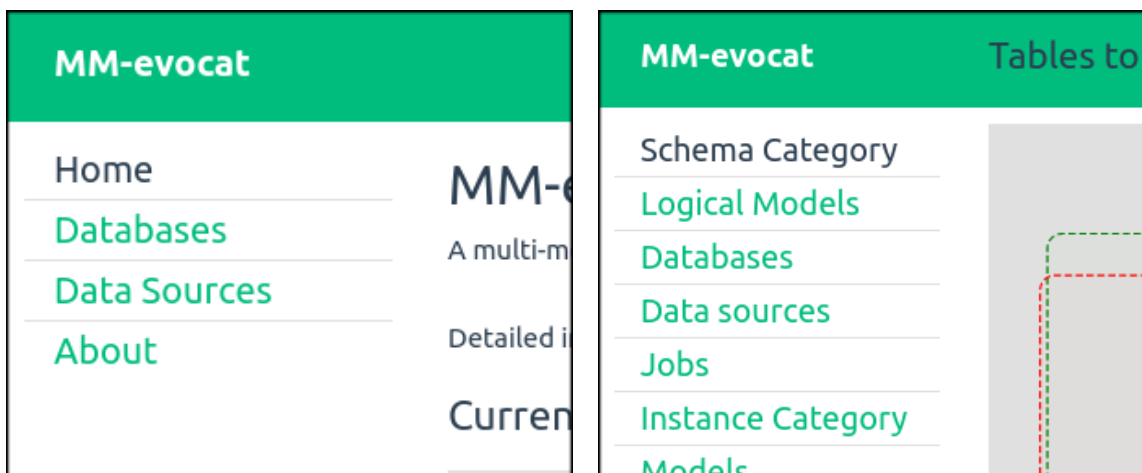
- na první pohled dobré, příjemné barvy, jen pár komponent na které mám klikat
- občas se vyskytují překryvy elementů (částečně když mám aplikaci na půl okna)
- překrytí nav menu



- přetéká okno modelování



- **ZAÚJALO MĚ, a trochu zmátlo...:** po rozkliknutí schémátka z hlavní stránky, se zobrazí jiné nav menu, to staré zmizí, u navů nejsem zvyklá na jejich překreslování v rámci aplikace (pokud se třeba nepřihlašuju), proto mi to ze začátku přišlo matoucí, než jsem si uvědomila, že je něco jinak
- název schéma se objeví v horní liště -> překresluje se zase statický prvek, název je pak odtržený, hledala bych ho spíš v levém navu (když už)



# Uživatelské hledisko - samotný nástroj

tlačítka

- Add object, Add morphism a Save - používat
- ostatní nedělají nic relevantního
- po zmáčknutí Save už nejde editovat existující věci !

inputy

- obsahující slovo "Iri" ignorovat

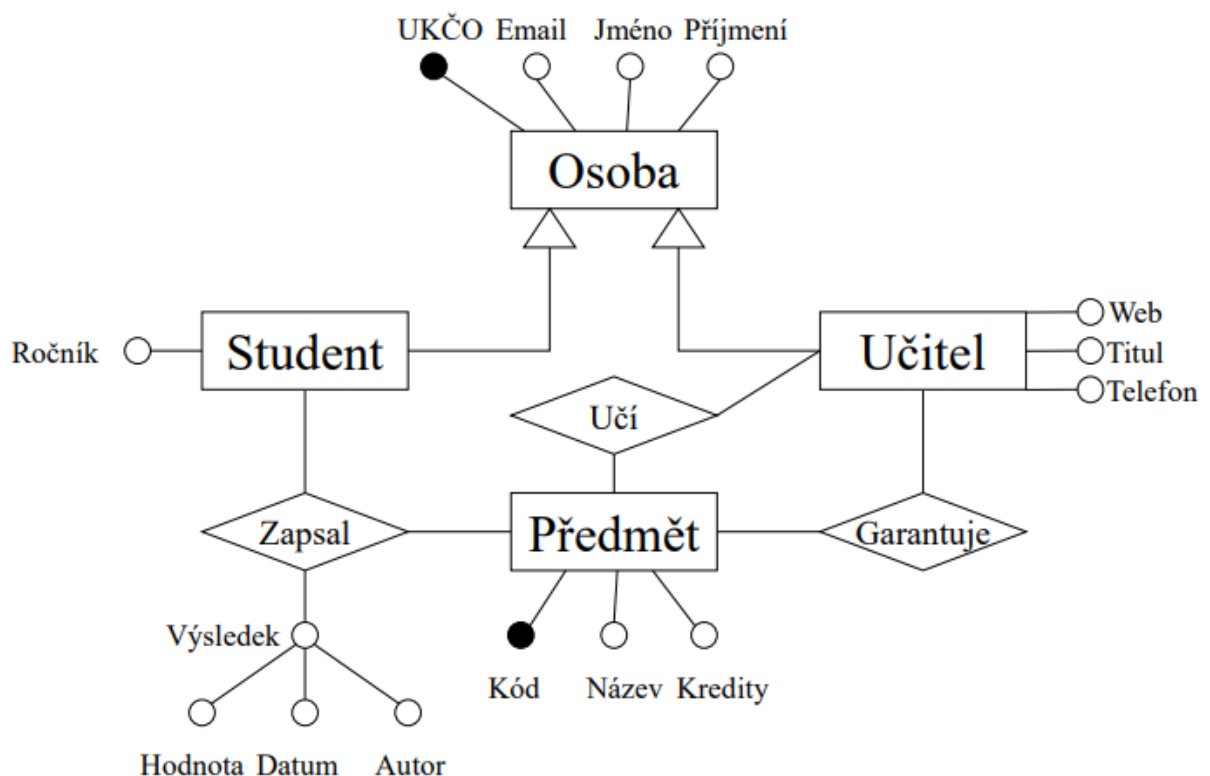
Ids (identifikátory)

- možnost přidat k objektům == přidávat !
- value - když mají nodes nějakou hodnotu
- ostatní co nejsou value, budou v sobě mít info o

## Poznámky k vytvoření schema category

### 1. Iterace

- vytvořila jsem schema "sarka - test sis" a převedla do něj ER model SISu

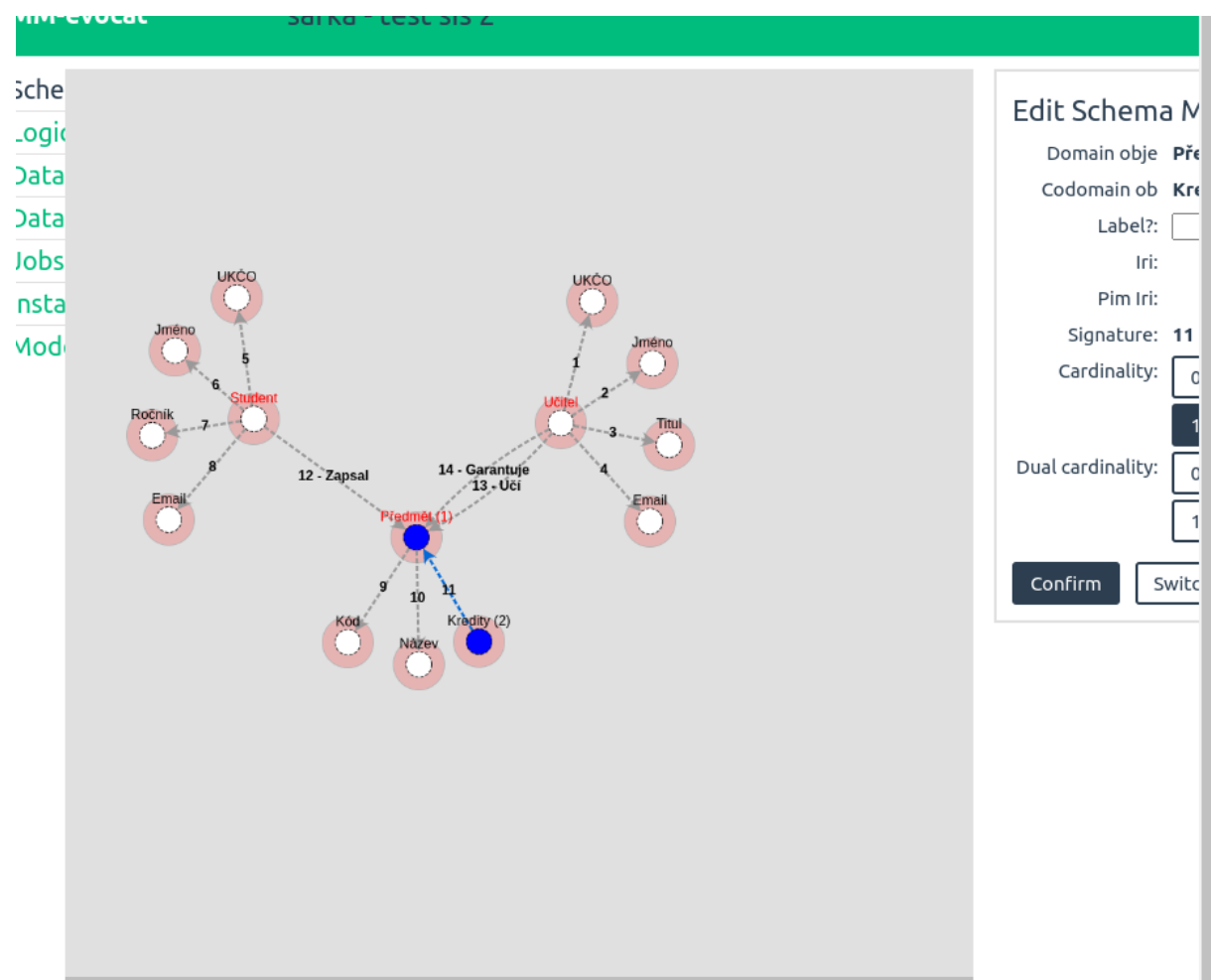


- provedení vytvořením Schema category bylo intuitivní (dělám krok po kroku) (když už jsem věděla, která tlačítka používat)
- na začátku jsem nevěděla, co přesně umožňuje koncepční schéma
- vše se dalo dořešit po konzultaci s databází nebo Jáchymem
- Zjistila jsem, že existují signatures, což je problém, uložila jsem si schéma a už ho nemůžu předělat, musím dělat znovu

- červený kruh kolem uzlů - když nemám node v databázi
- zabarvení labelu u objektu - když je vytvořen, má červený label
- signature - důležitý pro specifikaci morfismu, internal identifier
- každé mapování odpovídá jedné tabulce
- při používání mi trochu chyběly alt popisky, nebo varování ("chystáte se uložit, po této akci, už nebude možné upravit uložené části" a tak)

## 2. Iterace

- pročítám postupně dokumentaci a narážím na různé pojmy
  - Category Theory:
    - vytvoření kategorie (SK)
    - Identity morphism - pro každý objekt
    - *Base morphisms  $(f, g) \times$  Composite morphisms  $(f \circ g)$*
    - každý morfismus co není identita, existuje duál ( $f: A \rightarrow B$ ,  $f': B \rightarrow A$ )
    - conceptual schema = in this model schema category
    - root property?
- 2. iterace SK 'sarka - test sis 2'
  - už jsem rovnou naházela všechny nodes najednou, základy jdou rychleji
  - přidávání morfismů - **pořád se musím ujišťovat odkud kam je kardinalita**, představuji si to jako UML
  - teď se budu muset poprat s přidáváním ids, nemělo by to být těžké, schéma není nijak komplikované -> ok, dodělala jsem value ids a chtěla vyměnit jeden morfismus, jenže aplikace zamrzla a nestačila jsem nic uložit 😊



- moc se mi to nechce dělat celé od znova, tak se raději pokusím pochopit víc signatures z modelů co už v SKs jsou

**Evocat** - nástroj pro modelování a správu evoluce v multi-modelových datech (tj. struktura dat se může v čase měnit, aby např. vyhovovala novým uživatelským požadavkům, např. přidáme nebo přejmenujeme atribut, ale jsou zde i mnohem komplexnější operace)

cat = category theory

Root node?

Signatures tam jsou kvůli multi-modelovatelnosti?

Proč jsou skládané signatures značené odzadu po směru šipek? (Je to jako skládání fcí?)