Integrace relačních a grafových databází funkcionálně

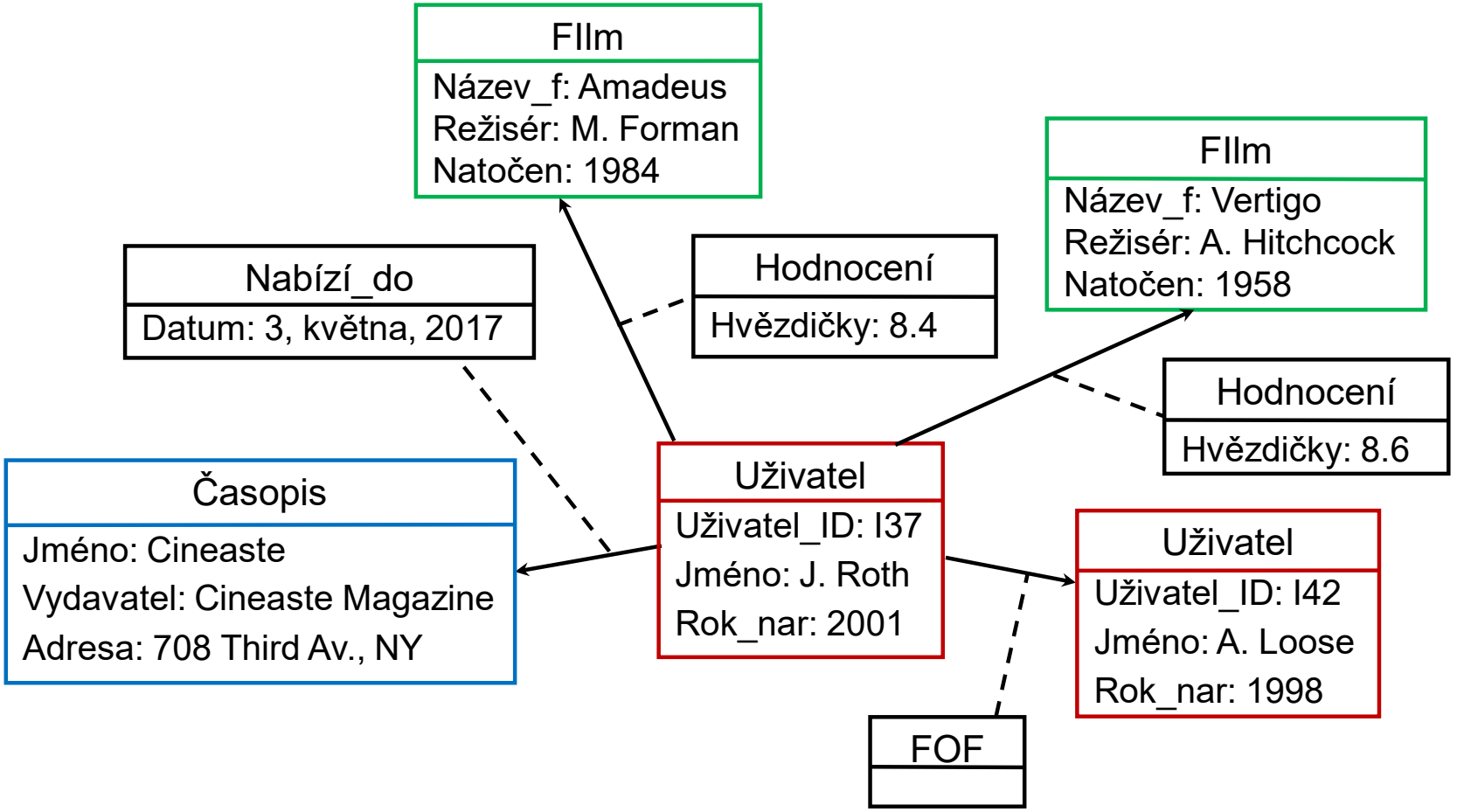
J. Pokorný
MFF UK, Praha

Obsah

- Úvod
- Funkcionální přístup k modelování dat
- Manipulace funkcí
  - jazyk (lambda) termů
  - dotazování nad relacemi a atributovými grafy
- Integrace relací a atributových grafů
- Závěry

Úvod

Příklad grafové databáze:



Úvod

Příklad relační databáze:

Herci(Jméno, Název_f, Role)

Filmy(Název_f, Natočen, Režisér, Žánr)

Jméno	Název_f	Role
Kim Novak F. M. Abraham Tom Hulce ...	Vertigo Amadeus Amadeus ...	Madeleine Salieri Mozart

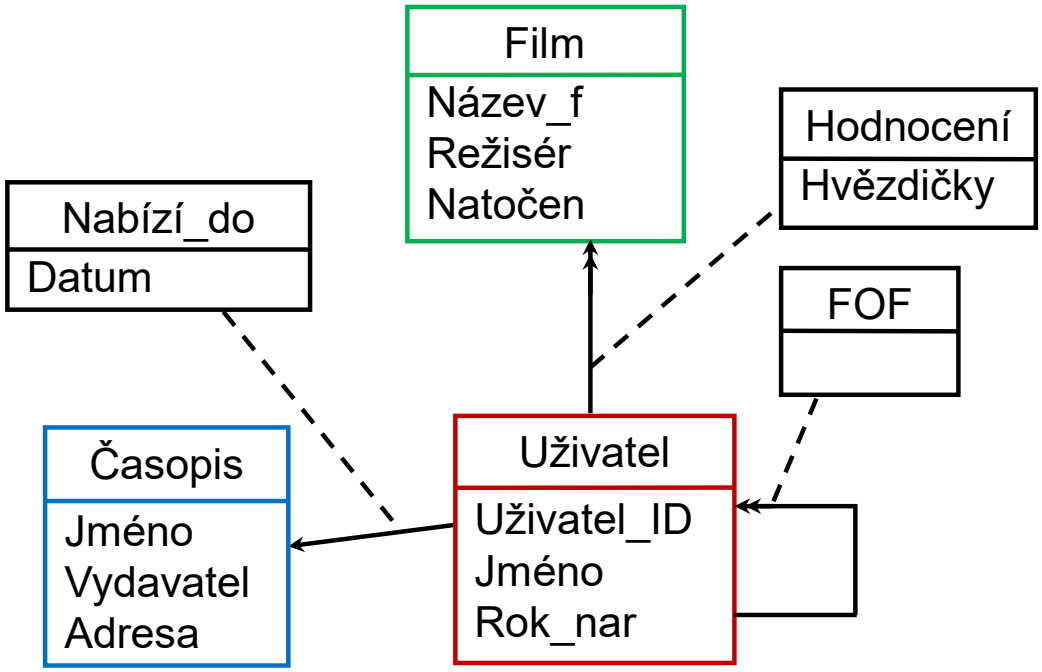
Název_f	Natočen	Režisér	Žánr
Vertigo Amadeus ...	1958 1984 ...	A. Hitchcock M. Forman ...	thriller drama ...

Úvod

- Terminologie: GDB, GSŘBD, GDBS (často pouze grafové databáze)
  - GDB může obsahovat jeden (velký) graf nebo
  - kolekci malých či středně velkých grafů
  - schémata grafové databáze většinou schází
- Problémy:
  - jak modelovat GDB,
  - jak dotazovat GDB,
  - jak integrovat GDB a nějaké jiné databáze
- Náš přístup
  - GDB je jeden graf
  - použijeme funkcionální schéma GDB a typovaný lambda kalkul jako prostředek dotazování
  - relace budou vyjádřeny také funkcionálně
  - integrace funkcionálně

Úvod

Příklad schématu GDB:



Grafový datový model

- (ohodnocený) atributový grafový datový model
  - entity (uzly)
  - vlastnosti (atributy)
  - značky (typy)
  - vztahy (hrany)
    - směr,
    - počáteční uzel,
    - koncový uzel
  - identifikátory

Entity a vztahy mohou mít vlastnosti, uzly a hrany mohou být označeny značkami. Jak uzly tak hrany jsou definovány jednoznačným identifikátorem.

- v pojmech teorie grafů: ohodnocené orientované atributové multigrafy
- mohou být použity i pro schémata GDB.

Funkcionální přístup k modelování dat

- Typované funkce vhodné pro modelování reálných datových objektů jsou atributy nahlížené jako empirické typované funkce popsané výrazem přirozeného jazyka.
- Předpokládáme bázi **B** … a množinu symbolů $S_1, S_2, \ldots, S_k$ ($k \geq 1$). Ty tvoří elementární typy.

  Jestliže S, R_1,..., R_n ($n \geq 1$) jsou typy, pak

  (i) $(S:T_1,\ldots,T_n)$ je funkcionální typ
  (ii) $(T_1,\ldots,T_n)$ je n-ticový typ

Množina typů *T* nad **B** je nejmenší množina obsahující typy z **B** a ty dané (i)-(ii).

Bool = {*TRUE*, *FALSE*} je také v **B**. Pak množiny a relace jsou $(Bool:T)$ resp. $(Bool:T_1,\ldots,T_n)$-objekty.

Funkcionální přístup k modelování dat

Jsou-li S_i z **B** interpretovány jako neprázdné množiny, pak (S:R_1,...,R_n) označuje množinu všech (totálních a parciálních) funkcí z R_1×...×R_n do S, (R_1,...,R_n) označuje kartézský součin R_1×...×R_n.

- aritmetické operace:
  - +, -, *, / jsou (*Number*:*Number*, *Number*)-objekty.
- logika:
  - **and**/(*BOOL*:*BOOL*, *BOOL*)-objekty,
  - univerzální R-kvatifikátor Π_R, a existenční R-kvantifikátor Σ_R jsou (*BOOL*:(*BOOL*:R))-objekty.
  - R-identita $=_R$ je (*BOOL*:*R*,*R*)-objekt.
- agregační funkce:

  $COUNT_S$/*Number*:(*Bool*:*S*), SUM/*Number*:(*Bool*:*Number*)

Funkcionální přístup k modelování dat

Příklad:

B = {*Uživatel*, *Film*, *U_ID*, *Jméno*, *Rok_nar*,…}.

pak např. výraz" filmy hodnocené uživatelem" označuje ((*Bool*:*Film*):*Uživatel*)-objekt, tj. (parciální) funkci *f*:*Uživatel*→(*Bool*:*Film*). Takové funkce reprezentují atributy.

- každá báze **B** sestává z deskriptivních a entitních typů.
- pro GDB můžeme pojímat typy entit jako množiny ID uzlů.
- typy *String*, *Number* atd., slouží pro domény vlastností.

GDB schéma funkcionálně

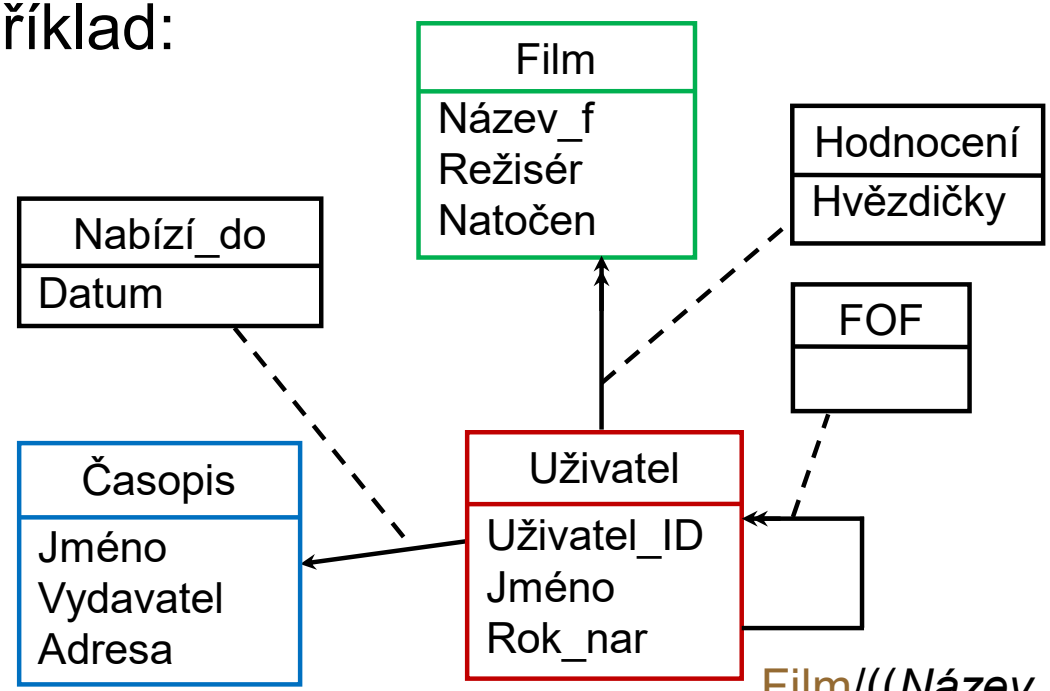
- Atributy typů: (*R*:*S*) a ((*Bool*:*R*):*S*), kde *R* a *S* jsou typy entit, tj. jednohodnotové a vícehodnotové atributy. Pro vlastnosti užíváme n-ticové atributy.
- Vlastnosti popisující typy entit jsou typu $((S_1,...,S_m):R)$ ($m \geq 1$), kde S_i jsou deskriptivní typy a *R* je entity typ. Vlastnosti hran jsou typu $((S_1,...,S_m,R_1):R_2)$ nebo $((Bool:S_1,...,S_m,R_1):R_2)$.

Příklad:

Film/((*Název_f*, *Režisér, Natočen*):*Film*)
Uživatel/((*U_ID*, *Jméno*, *Rok_nar*):*Uživatel*)
Časopis/((*Jméno*, *Adresa*, *Vydavatel*):*Časopis*)
FOF/((*Bool*:*Uživatel*):*Uživatel*)
Hodnocení/((*Bool*:*Hvězdičky*, *Film*):*Uživatel*)
Nabízí_do/((*Datum*, *Časopis*):*Uživatel*)

GDB schéma funkcionálně

Příklad:



Film/((*Název_f*, *Režisér, Natočen*):*Film*)
Uživatel/((*U_ID*, *Jméno*, *Rok_nar*):*Uživatel*)
Časopis/((*Jméno*, *Adresa*, *Vydavatel*):*Časopis*)
FOF/((*Bool*:*Uživatel*):*Uživatel*)
Hodnocení/((*Bool*:*Hvězdičky*, *Film*):*Uživatel*)
Nabízí_do/((*Datum*, *Časopis*):*Uživatel*)

Relační schéma funkcionálně

(Relační) atributy A_i:D_j budou použity jako S_i. S_i jsou neprázdné množiny hodnot, $S_i \neq S_j$ pro $i \neq j$. Pak relace jsou (*Bool*:S_1, …,S_n)-objekty.

Příklad: relace

Herci(Jméno, Název_f, Role)

Filmy(Název_f, Natočen, Režisér, Žánr),

tj., atributy

Herci/(*Bool*:*Jméno*, *Název_f*, *Role*),

Filmy/(*Bool*:*Název_f*, *Natočen*, *Režisér*, *Žánr*).

Poznámka: primární klíče nejsou uvažovány.

Manipulace funkcí

Jazyk (lambda) termů LT

- Konstanty a proměnné pro každý typ z **T**.
- Nechť typy R, S, R_1,..., R_n (n ≥1) jsou z **T**.

(1) Každá proměnná typu R is term typu R.

(2) Každá konstanta typu R je term typu R.

(3) Jestliže M je term typu (S:R_1,...,R_n), a N_1,...,N_n jsou termy typů R_1,...,R_n, pak $M(N_1,...,N_n)$ je term typu S.
/aplikace/

(4) Jestliže x_1,...,x_n jsou různé proměnné typů R_1,...,R_n a M je term typu S, pak $\lambda x_1,...,x_n(M)$ je term typu (S:R_1,...,R_n).
/lambda abstrakce/

(5) Jestliže N_1,...,N_n jsou termy typů R_1,...,R_n, pak
(N_1,...,N_n) je term typu (R_1,...,R_n). /n-ticový/

(6) Jestliže M je term typu (R_1,...,R_n), pak M[1],...,M[n] jsou termy typů R_1,...,R_n.
/komponenty/

Manipulace funkcí

LT stručněji:

(1) x	proměnné
(2) f	konstanty (funkce)
(3) $M(N_1,...,N_n)$	aplikace
(4) $\lambda x_1,...,x_n(M$	lambda abstrakce
(5) $(N_1,...,N_n)$	n-tice
(6) $M[1],...,M[n]$	komponenty n-tice

Dotazování nad relacemi a atributovými grafy

- GDB: „Najdi názvy filmů režírovaných Spielbergem", může být vyjádřen termem

  $\lambda\, t\ (\exists\, m, r\ \mathrm{Film}(m)(t, \text{'Spielberg'}, r)\ )$

- RDB: "Najdi herce, kteří hrají v každém filmu režiséra Spielberga"

  $\lambda\, n\ (\forall t\ (\exists\, re, g\ \mathrm{Filmy}(t, re, \text{'Spielberg'}, g)\ \textbf{implies}\ \exists\, ro\ \mathrm{Herci}(n, t, ro)))$

Více syntaktického „cukru“:

$\{t^{Název\_f} \mid \textbf{exists}\ m^{Film}\ \mathrm{Film}(m^{FIlm})(t^{Název\_f}, \text{'Spielberg'}^{Režisér})\}$

$\{n^{Jméno} \mid \textbf{foreach}\ t^{Název\_f}\ (\mathrm{Filmy}(t^{Název\_f}, \text{'Spielberg'}^{Režisér})\ \textbf{implies}\ \mathrm{Herci}(n^{Jméno}, t^{Název\_f}))\}$

Obecněji:

- eliminace některých existenčních kvantifikátorů a proměnných,
- explicitní typování proměnných a konstant,
- komponenty jménem: místo Film(Id4)[1] čtivěji Film(Id4).Název_f

Integrace relací a atributových grafů

Tři způsoby integrace relačních a NoSQL databází: nativní, hybridní, a redukující se k jedné volbě (buď relační nebo NoSQL).

Vyvíjejí se přístupy :

- polyglotní persistence,
- vícemodelový přístup,
- víceúrovňové modelování,
- konverze schématu a dat.

Víceúrovňové modelování pokrývá následující přístupy:

(a) speciální abstraktní model,

(b) NoSQL-on-RDBMS,

(c) integrace ontologií.

Integrace relací a atributových grafů

Tři způsoby integrace relačních a NoSQL databází: nativní, hybridní, a redukující se k jedné volbě (buď relační nebo NoSQL).

Vyvíjejí se přístupy:

- polyglot persistence,
- vícemodelový přístup,
- víceúrovňové modelování,
- konverze schématu a dat.

relevantní zde

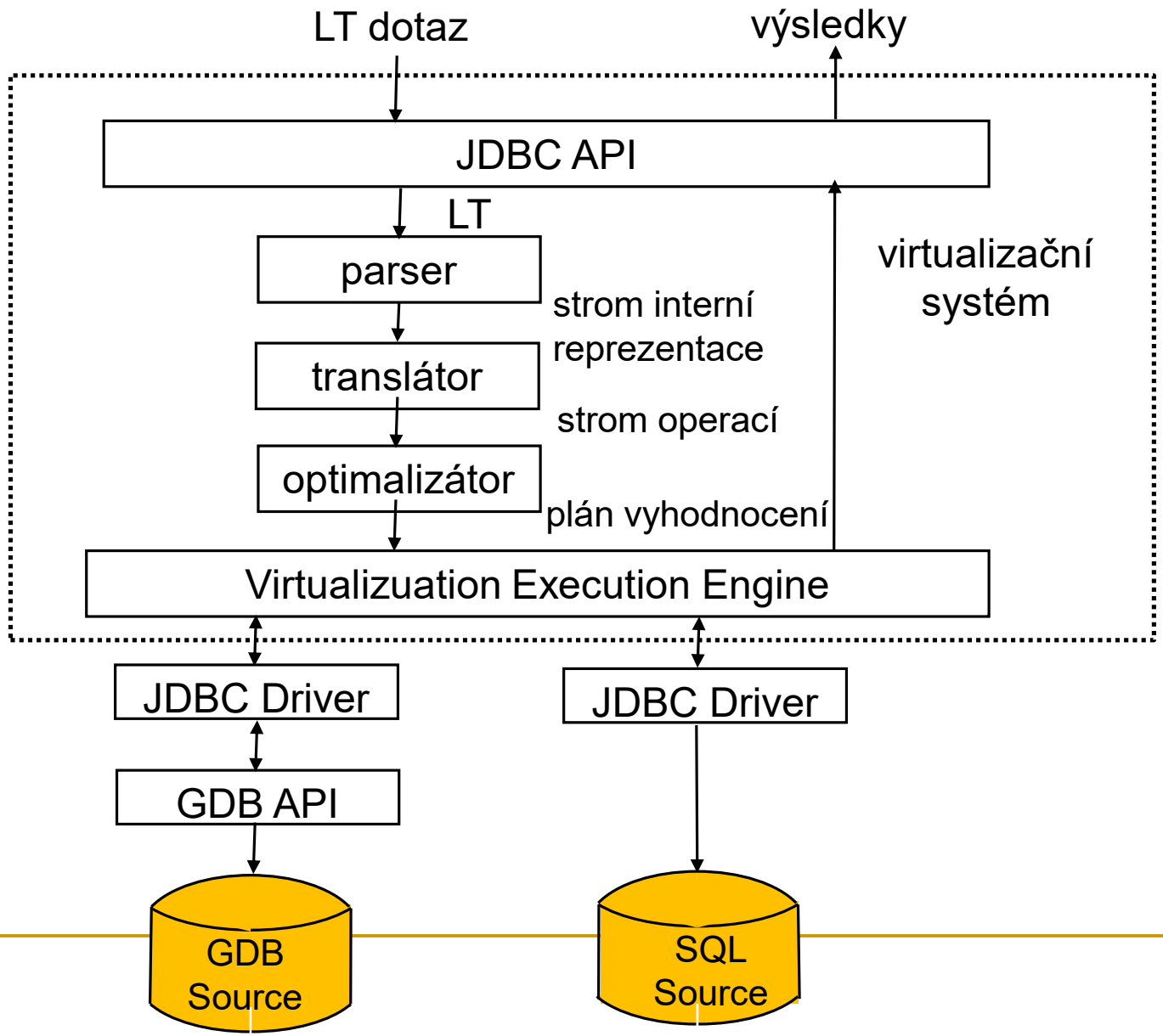
Víceúrovňové modelování pokrývá následující přístupy:

- (a) speciální abstraktní model,
- (b) NoSQL-on-RDBMS,
- (c) integrace ontologií.

relevantní zde

Př.: Speciální abstraktní model NoAM (NoSQL abstraktní Model) (Bugiotti, et al, 2014).

Integrace relací a atributových grafů



Integrace relací a atributových grafů

Problém: na datové úrovni jsou příslušné databáze většinou heterogenní:

- Elementární typ *Název_f* atributu Film nemá stejnou doménu jako dom(Název_f) relace Filmy. Předpokládat můžeme pouze jejich neprázdný průnik.
- Pro integraci obou schémat můžeme použít přejmenování některých atributů a/nebo relací (zde Film vs. Filmy).

Integrace relací a atributových grafů

Příklad dotazu nad integrovanou GDB a RDB:

$$\lambda\, u^{Uživatel}, g^{Žánr}, n^{Number}$$
$$(n^{Number}=\mathrm{COUNT}_{Movie}\ (\lambda\, m^{Film}\ (\mathrm{Hodnocení}(u^{Uživatel})(m^{Movie})\ \mathbf{a}$$
$$\exists\ t^{Název\_f}\ s^{Název\_f}\ \mathrm{Film}(m^{Film}).t^{Název\_f} = s^{Název\_f}\ \mathbf{and}$$
$$\mathrm{Filmy}(s^{Název\_f}, g^{Žánr})$$
$$)$$
$$)$$
$$)$$

vyjadřuje dotaz “Najdi pro každého uživatele a žánr počet recenzí, které v tomto žánru udělal”.

Závěry a náměty do budoucna

Současné výzvy pro databázový výzkum této infrastruktury zahrnují:

- Nalezení vhodné a dostatečně silné podmnožiny LT pokrývající dotazovací požadavky na GDB integrovanou s RDB,
- vyvinout smysluplnou a užitečnou uživatelskou verzi dotazovacího jazyka založeného na LT,
- vyvinout prototype používající SQL engine a Neo4j GSŘBD pro zdrojové databáze.