

Dokumentácia k inžinierskemu dielu

*Tvorba "ľahko" sémantického obsahu pre adaptívny webový
(výučbový) portál*

Tímový projekt

tím č. 13 – Thirteam

Thirteam

Bc. Martin Franta
Bc. Martin Gajdoš
Bc. Martin Habdák
Bc. Róbert Kocian
Bc. Petra Vrablecová
Bc. Zuzana Zimová

Vedúci pedagóg: Ing. Marián Šimko, PhD.

e-mail: tim13@googlegroups.com

Ak. rok : 2011/2011

Obsah

1	Úvod	1-1
1.1	Použité pojmy	1-1
2	Opis prototypu	2-1
2.1	Funkcionalita systému	2-1
2.1.1	<i>Home</i>	2-1
2.1.2	<i>Repo Detail (RD2)</i>	2-2
2.1.3	<i>Metadata Detail – table (V1)</i>	2-3
2.1.4	<i>Metadata Detail – graph (V1)</i>	2-4
2.1.5	<i>Repo Detail (RDVI)</i>	2-5
2.1.6	<i>Document Detail – edit (DE1)</i>	2-6
2.1.7	<i>Document Detail – view (DVI)</i>	2-6
2.2	Dátový model systému.....	2-8
2.2.1	<i>Document</i>	2-8
2.2.2	<i>RDT</i>	2-8
2.2.3	<i>Repository</i>	2-8
2.2.4	<i>User</i>	2-8
2.2.5	<i>Annotation</i>	2-9
2.2.6	<i>Variant</i>	2-9
2.3	Architektúra systému	2-11
2.4	Systémové komponenty	2-12
2.4.1	<i>Prihlásenie používateľa</i>	2-12
2.4.2	<i>Komunikácia s klientmi</i>	2-13
2.4.3	<i>Dokumenty</i>	2-16
	Vytvorenie a úprava dokumentu	2-17
	Repozitáre	2-17
	Presúvanie dokumentov	2-17
	Import repozitára a metadát.....	2-18
	História importu	2-20
	Export repozitára	2-21
	Validácia dokumentu.....	2-21
	Kontrola integrity dokumentu	2-22

View a preview dokumentu	2-23
Verziovane.....	2-24
Vzťahy medzi RDT a dokumentmi	2-25
Vzťahy medzi dokumentmi	2-26
2.4.4 Anotácie	2-27
Reprezentácia anotácií.....	2-27
Anotácie a anotačný pásik	2-27
Riešenie konfliktov.....	2-27
2.4.5 Metadáta.....	2-28
Varianty	2-28
Vzťahy medzi metadátami a repozitármi	2-28
RDT a vzťahy medzi RDT	2-29
Graf.....	2-29
Príloha A: Obrazovky	A-1
A.1 Home	A-1
A.2 Repo Detail (RD2)	A-1
A.3 Metadata Detail – table (V1).....	A-1
A.4 Metadata Detail – graph (V1)	A-1
A.5 Repo Detail (RDV1)	A-1
A.6 Document Detail – edit (DE1)	A-1
A.7 Document Detail – view (DV1)	A-1
A.8 Tabuľky v systéme	A-9

1 Úvod

Témou nášho tímového projektu je *Tvorba "lahko" sémantického obsahu pre adaptívny webový (výučbový) portál*. Cieľom projektu je vytvoriť systém správy obsahu pre adaptívny webový portál a využiť tento systém pre správu výučbového obsahu systému ALEF. Výsledný projekt bude môcť byť používaný pedagógmi, ktorí prostredníctvom neho budú môcť vytvárať učebné texty s využitím metadát.

Vo všeobecnosti by mal byť projekt využiteľný pre rôzne typy adaptívnych webových portálov, ktoré využívajú sémantické metadáta, vzťahy medzi dokumentmi a na nich založené odporúčania.

Implementácia projektu je vytvorená v jazyku Ruby a frameworku Ruby on Rails, použitom aj pri vývoji systému ALEF. Zároveň použitie Ruby on Rails napomáha organizovaniu zdrojových kódov a jednoduchšej údržbe, čo bude výhodou pre ďalšie tímy, ktoré tento systém budú spravovať a vylepšovať.

Tím vyvíja projekt podľa metodiky SCRUM.

1.1 Použité pojmy

- Product backlog* zoznam požiadaviek zákazníka vytvorený na začiatku šprintu – obsahuje vlastnosti, ktoré by mal systém obsahovať po ukončení šprintu
- Dokument* obsahová entita, v prípade systému ALEF predstavuje XML dokument s výučbovým obsahom (tzv. *learning object*)
- Repozitár* úložisko dokumentov – priečinok s dokumentmi rovnakej témy, v prípade systému ALEF predstavuje jeden *kurz*
- RDT* *relevant domain term* – kľúčové slovo alebo viacslovný výraz vzťahujúci sa na tému resp. zameranie dokumentu

2 Opis prototypu

2.1 Funkcionalita systému

Systém sa skladá zo siedmich hlavných obrazoviek (Príloha A: Obrazovky). Každá z nich poskytuje špecifickú funkcionálnosť pre správu obsahu adaptívneho portálu. Funkcie sú podrobnejšie opísané v nasledujúcich podkapitolách.

2.1.1 Home

Úvodná obrazovka systému (Príloha A: Obr. 16) obsahuje dve inteligentné tabuľky, ktoré reprezentujú prehľad repozitárov a variantov metadát, pričom pri tabuľke repozitára je možné importovať a exportovať celé databázy učebných textov aj s variantmi.

Export

Export je naviazaný na tabuľku repozitárov, tak aby bolo možné vyexportovať vybraný repozitár v tabuľke.

Import

Pri importe súborov systém rozbalí daný súbor typu zip a zobrazí prehľad položiek, ktoré tento súbor obsahuje, pričom je možné medzi týmito položkami vyberať, či ich chceme uložiť, a ktoré nie. Pri importe je možné zvoliť validáciu importu. Načítanie súboru prebieha na pozadí stránky, takže nie je nutné aby sa stránka znova načítala.

Repozitáre

Repozitáre sú v úvodnej obrazovke reprezentované tabuľkou, ktorú je možné scrolovať, filtrovať, či už komplexne alebo po stĺpcoch a usporiadať podľa stĺpcov. Akcie pri tejto tabuľke sú buď riadkové alebo pre celú tabuľku. Pre celú tabuľku je možné pridávať nové repozitáre a exportovať vybrané a pri riadkových akciách je možné publikovať, robiť klasické operácie ako zobrazenie, editovanie a mazanie zvoleného repozitára.

Varianty

Varianty majú rovnaké riadkové a komplexné operácie ako repozitáre v tabuľke repozitárov s tým rozdielom, že akcie a tabuľka je prispôbená na prácu s variantmi

The screenshot shows the COME2J Home page. At the top right, there is a user profile 'm@m.sk' and a 'Sign out' link, labeled '1'. Below this is the 'Repositories' section, labeled '2', which contains a table with columns: Name, Version, HasChanged, #Documents, and Actions. The table lists three repositories: 'Lisp 2010', 'Lisp 2011', and 'prolog'. Below the table are three buttons: 'Add' (labeled '3'), 'Export selected' (labeled '4'), and 'Import repository' (labeled '5'). Below the 'Repositories' section is the 'Variants' section, labeled '6', which contains a table with columns: Name, Version, HasChanged, #RDTs, #RDT relationships, #Repositories, and Actions. The table lists three variants: 'course1_variant2', 'course2_variant2', and 'Lisp variant'. Below the 'Variants' section is an 'Add' button, labeled '7'.

Obr. 1. Obrazovka Home.

Legenda

1. Prihlásený používateľ
2. Zoznam repozitárov
3. Pridanie nového repozitára
4. Export vybraných repozitárov
5. Import nových dát (repozitáre, varianty) do systému
6. Zoznam variantov
7. Pridanie nového variantu

2.1.2 Repo Detail (RD2)

Pri zobrazení konkrétneho repozitára (Príloha A: Obr. 17) sa dostávame do detailu repozitára. Táto časť systému obsahuje varianty metadát, ktoré sú priradené k zobrazenému repozitáru a tiež dokumenty, ktoré tento repozitár obsahuje.

Dokumenty

Pri dokumentoch je dôležitou funkcionalitou presun dokumentu do iného repozitára. Princíp fungovania spočíva vo vybrání dokumentu, ktorý chceme presunúť v tabuľke, a vo vybrání repozitára v Pop-Upe.

Prislúchajúce varianty

Priradené varianty metadát majú všetky ostatné riadkové akcie rovnaké ako iné tabuľky v systéme, pričom zaujímavé je samotné priradenie variantu k repozitáru.

K repozitáru je možné priradiť viacero variantov. Túto akciu je možné zabezpečiť aj bez toho, aby sa stránka znova načítala, pričom túto funkciu zabezpečuje AJAX.

2.1.3 Metadata Detail – table (V1)

Obrazovka V1 (Príloha A: Obr. 18) slúži na prácu s variantom metadát. Variant je zobrazený vo forme dvoch tabuliek – tabuľka so zoznamom termov (RDT) a tabuľka so zoznamom vzťahov medzi RDT. Obrazovka poskytuje nasledujúcu funkcionálnosť:

Pridanie RDT

Pod zoznamom RDT sa nachádza pole pre vyplnenie názvu termu. Po stlačení tlačidla Add sa vytvorí nové RDT so zadaným názvom a pridá sa do zoznamu. V jednom variante metadát sa nemôžu nachádzať dve RDT s rovnakým názvom.

Úprava, vymazanie RDT

V zozname RDT sa vedľa každej položky nachádzajú akcie pre úpravu a vymazanie termu.

Po spustení akcie pre úpravu termu kliknutím na danú ikonu sa zobrazí riadkový editor pre dané RDT v zozname termov. Vtedy je možné meniť v poli názov RDT. Vedľa polia sa nachádzajú akcie pre uloženie alebo zrušenie úpravy termu.

Po kliknutí na ikonu pre vymazanie termu sa objaví dialóg pre potvrdenie voľby. V prípade, že sa RDT nachádza v nejakých vzťahoch, je používateľ najskôr vyzvaný vymazať dané vzťahy.

Pridanie vzťahu

Pod zoznamom vzťahov sa nachádza formulár pre pridanie nového vzťahu. Obsahuje polia pre dve RDT, medzi ktorými sa má vytvoriť vzťah, pole pre typ vzťahu a pole pre zadanie váhy vzťahu. Polia pre RDT môžu obsahovať iba hodnotu zo zoznamu RDT a táto hodnota nesmie byť v oboch poliach rovnaká. Typy vzťahov sú z pevne zadefinovanej množiny typov vzťahov. Váha vzťahu musí byť z intervalu $(0,1>$. Po stlačení tlačidla Add sa vytvorí vzťah so zadanými atribútmi a pridá sa do zoznamu vzťahov.

Úprava, vymazanie vzťahu

Vedľa každej položky v zozname vzťahov sa nachádzajú akcie pre úpravu a vymazanie vzťahu.

Po spustení akcie pre úpravu vzťahu sa v danom riadku zobrazí riadkový editor pre vzťah. Vedľa polí pre úpravu vzťahu sa nachádzajú akcie pre uloženie vzťahu alebo zrušenie upravovania.

Po kliknutí na ikonu pre vymazanie vzťahu sa zobrazí potvrdzujúci dialóg. Po jeho potvrdení sa vzťah vymaže.

Vymazanie zvolených RDT alebo vzťahov

Oba zoznamy majú možnosť označiť viacero položiek naraz a vykonať nad nimi hromadnú akciu vymazania položiek.

Zoradovanie a vyhľadávanie

Oba zoznamy majú možnosť zoradovania a vyhľadávania položiek podľa stĺpcov.

Filtrovanie vzťahov podľa zvoleného RDT

Po kliknutí na položku zo zoznamu RDT sú v zozname vzťahov vyfiltrované práve tie vzťahy, v ktorých vystupuje zvolené RDT.

Prepnutie na graf

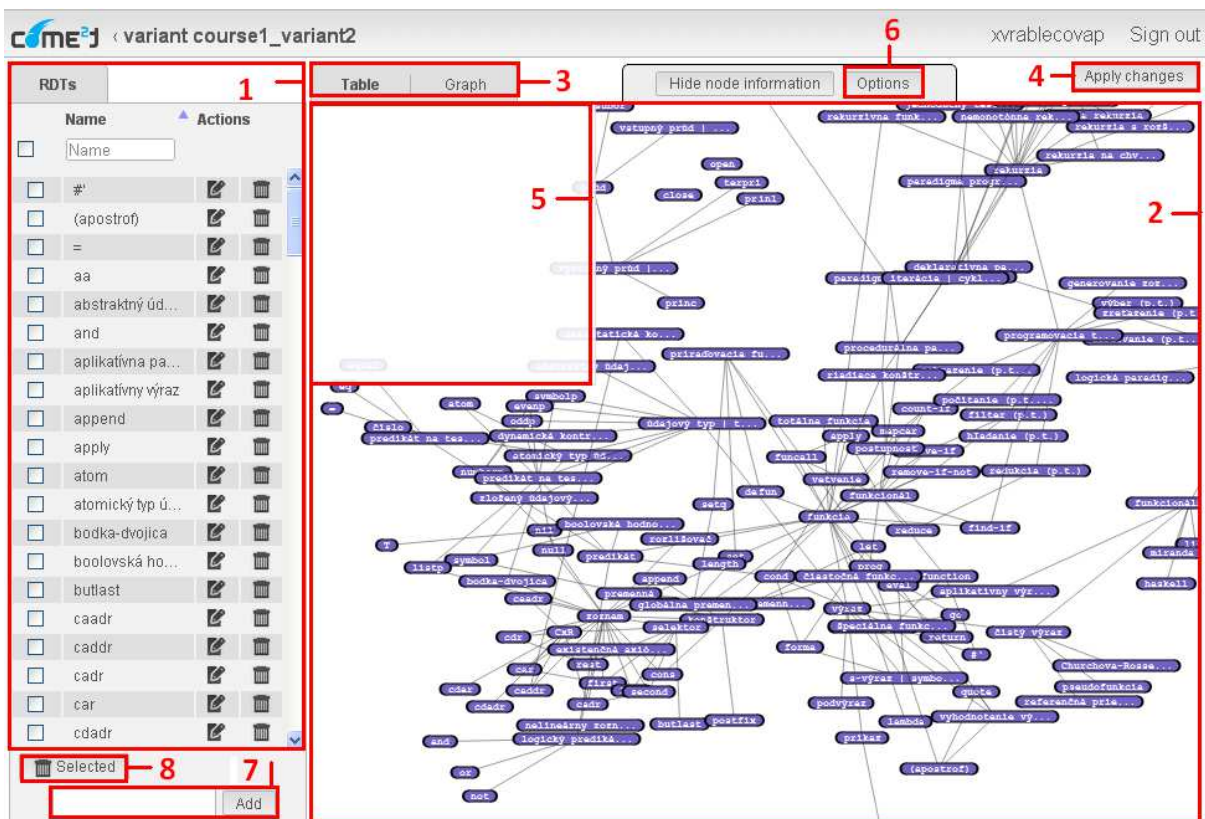
Z tejto obrazovky sa dá po kliknutí na tlačidlo Graph prepnúť do grafového zobrazenia variantu metadát. Funkcionalita tejto obrazovky je opísaná ďalej.

Uloženie zmien

Obrazovka obsahuje tlačidlo Apply changes. Po jeho stlačení sa uložia všetky vykonané zmeny do databázy. Dovtedy všetky úpravy prebiehajú len na klientovi. Pri pokuse opustiť obrazovku je používateľ vyzvaný k uloženiu vykonaných zmien.

2.1.4 Metadata Detail – graph (V1)

Táto obrazovka (Príloha A: Obr. 19) slúži na manipuláciu s metadátami vo forme grafu. Poskytuje väčšinu funkcionality predchádzajúcej obrazovky, ale pomocou iných ovládacích prvkov.



Obr. 2. Obrazovka Metadata Detail - graph (V1).

Legenda

1. Zoznam RDT prislúchajúcich k repozitáru
2. Zobrazenie grafovej vizualizácie metadát
3. Prepínanie medzi tabuľkovým a grafovým zobrazením
4. Aplikovanie vykonaných zmien do systému

5. Zoznam susedov prislúchajúcich k vrcholu
6. Možnosti zobrazenia grafu
7. Pridanie nového RDT
8. Vymazanie zvolených RDT

2.1.5 Repo Detail (RDV1)

Obrazovka RDV1 (Príloha A: Obr. 20) poskytuje funkcie pre vytváranie vzťahov medzi dokumentmi a RDT z aktuálneho variantu. Obsahuje zoznam dokumentov daného repozitára. Na pravej strane sa nachádza zoznam priradených RDT pre aktuálne zvolený dokument. Obrazovka poskytuje nasledujúcu funkcionálnosť:

Zobrazenie zoznamu priradených RDT pre zvolený dokument

Po zvolení dokumentu zo zoznamu dokumentov sa na pravej strane zobrazí zoznam vzťahov s priradenými RDT. Každý vzťah má atribúty typ vzťahu z pevne definovanej množiny typov vzťahov a váhu vzťahu z intervalu $(0,1>$.

Vytvorenie vzťahu medzi dokumentom a RDT

Pod zoznamom s priradenými RDT k vybranému dokumentu sa nachádza formulár pre vytvorenie nového vzťahu medzi vybraným dokumentom a zadaným RDT. RDT musí patriť do aktuálneho variantu metadát. Po stlačení tlačidla Add sa vytvorí vzťah so zadanými atribútmi a pridá sa do zoznamu vzťahov s RDT vybraného dokumentu.

Úprava, vymazanie vzťahu medzi dokumentom a RDT

Vedľa každej položky so zoznamu vzťahov s priradenými RDT vybraného dokumentu sa nachádzajú akcie pre úpravu vzťahu a vymazanie vzťahu.

Po spustení akcie pre úpravu vzťahu sa v danom riadku zobrazí riadkový editor pre vzťah. Vedľa polí pre úpravu vzťahu sa nachádzajú akcie pre uloženie vzťahu alebo zrušenie upravovania.

Po kliknutí na ikonu pre vymazanie vzťahu sa zobrazí potvrdzujúci dialóg. Po jeho potvrdení sa vzťah vymaže.

Vymazanie zvolených vzťahov

Zoznam s priradenými RDT pre daný dokument obsahuje možnosť označiť viacero vzťahov a vykonať nad nimi hromadnú akciu vymazania vzťahov.

Náhľad dokumentu

Vedľa každej položky zo zoznamu dokumentov repozitára sa nachádza ikona oko. Po nastavení kurzora na túto ikonu sa zobrazí náhľad dokumentu vo forme tooltipu.

Zobrazenie dokumentu

V každom riadku zoznamu dokumentov z daného repozitára sa nachádza odkaz na zobrazenie dokumentu, ktorý prenesie používateľa na obrazovku DV1, ktorej funkcionálnosť je opísaná nižšie.

Zoradovanie a vyhľadávanie

2 Opis prototypu

Oba zoznamy – zoznam s dokumentmi repozitára aj zoznam so vzťahmi s priradenými RDT vybratého dokumentu je možné zoraďovať a vyhľadávať v nich podľa stĺpcov.

2.1.6 Document Detail – edit (DE1)

Obrazovka DE1 (Príloha A: Obr. 21) slúži na editovanie XML dokumentov.



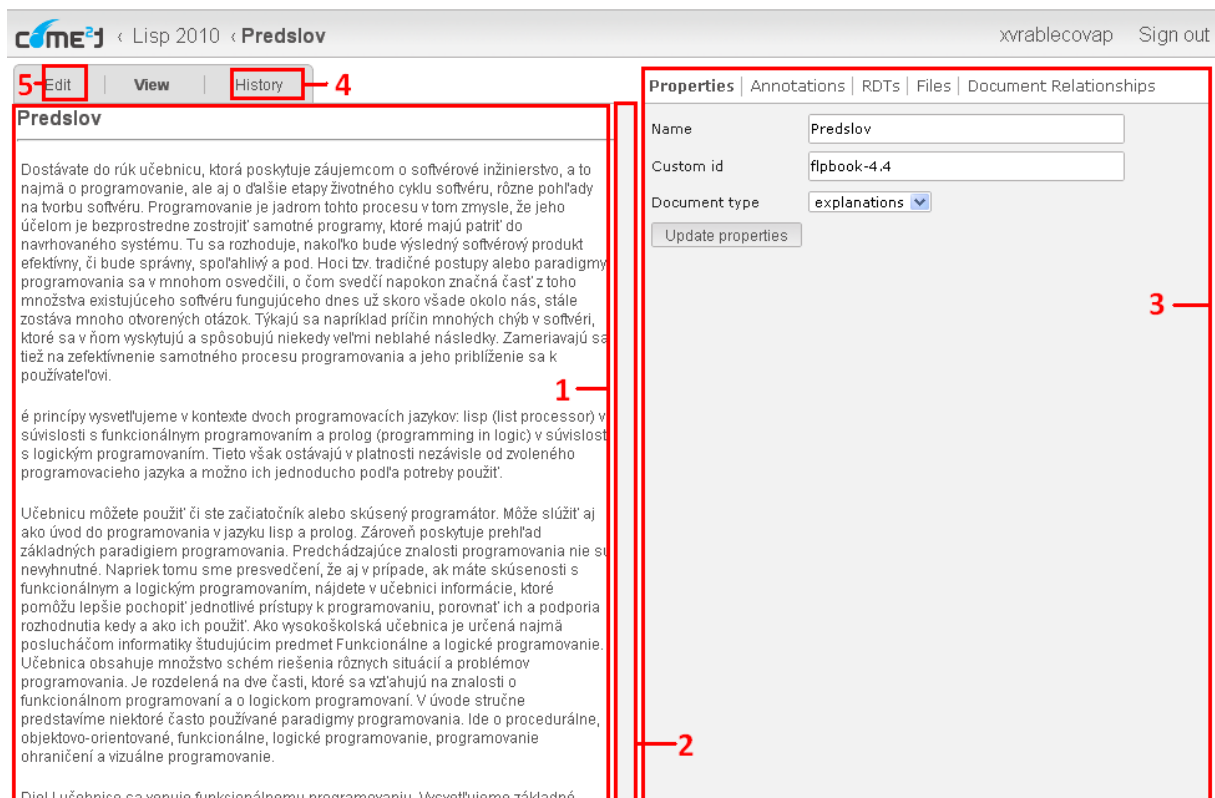
Obr. 3. Obrazovka Document Detail (DE1).

Legenda

1. XML editor
2. Možnosti editora – zalamovanie riadkov, rozťahnutie na celú obrazovku
3. Možnosti pre zapnutie validácie XML, kontroly integrity dokumentu, uloženie dokumentu, náhľad dokumentu
4. Formulár pre priloženie súboru k dokumentu
5. Tlačidlo pre zobrazenie/skrytie príbuzných entít dokumentu (vlastnosti dokumentu, RDT, anotácie, súbory, vzťahy k ostatným dokumentom)
6. Zobrazenie histórie dokumentu, možnosť návratu k staršej verzii dokumentu

2.1.7 Document Detail – view (DV1)

Obrazovka DV1 (Príloha A: Obr. 22) slúži na prezeranie dokumentov, ale najmä na prácu s anotáciami.



Obr. 4. Obrazovka Document Detail (DV1).

Legenda

1. Zobrazenie obsahu dokumentu
2. Anotačný pásik
3. Príbuzné entity dokumentu (vlastnosti dokumentu, RDT, anotácie, súbory, vzťahy k ostatným dokumentom)
4. Zobrazenie histórie dokumentu, možnosť návratu k staršej verzii dokumentu
5. Odkaz na obrazovku editovania dokumentu

2.2 Dátový model systému

Dátový model sa nachádza na Obr. 5. Zobrazuje entity existujúce v systéme a ich vzájomné vzťahy. Popis modelu je rozdelený na štyri hlavné časti.

2.2.1 Document

Learning Object je v systéme chápaný ako dokument. Reprezentuje ho entita *Document*. Dokument je vždy len jedného typu. Patrí do jedného adresára. K dokumentu sa priradujú viaceré súbory, ktoré sú implementované pomocou entity *Asset*. Dokument môže mať priradených viacero súborov.

Medzi dokumentmi môže existovať vzťah. Definuje ho entita *DocumentRelationship*. Vzťah má jeden typ a patrí do jedného variantu. Na rozdiel od ostatných vzťahov nemá žiadnu váhu.

Tvorcom dokumentu je používateľ systému, preto má dokument vzťah s entitou *User*. Používateľ je tvorcom viacerých dokumentov. Dokument má vždy len jedného autora. Je ním chápaný používateľ, ktorý ho na začiatku vytvorí.

2.2.2 RDT

Entita *RDT* predstavuje *Relevant Domain Term*. Je definovaná názvom a variantom, do ktorého patrí. Entita *Variant* je definovaná názvom. Medzi jednotlivými RDT existujú vzťahy, ktoré spolu s RDT slúžia na reprezentovanie ľahkej sémantiky. Rozdeľujeme ich na:

- *RDT-RDT* - vzťah medzi dvomi RDT. Je reprezentovaný entitou *RdtsRelationship*. Každý vzťah má určitú váhu. Jej hodnota patrí do intervalu $(0,1>$. Typ vzťahu je určuje entita *RdtsRelationshipType*.
- *RDT-Document* - vzťah medzi RDT a dokumentom. Reprezentovaný entitou *RdtDocumentRelationship*. Váha vzťahu je číslo z intervalu $(0,1>$. Typ vzťahu určuje entita *RdtDocumentRelationshipType*. Vzťahy medzi RDT a dokumentom nepatria do variantu.

Vzťah definuje vždy prepojenie len medzi dvoma RDT alebo medzi jedným RDT a jedným dokumentom. Vzťah má vždy len jeden typ, pričom viacero z nich môže byť toho istého typu.

2.2.3 Repository

Entita *Repository* predstavuje repozitár obsahujúci dokumenty. Do repozitára patrí viacero dokumentov. Neobsahuje dva dokumenty s rovnakým názvom. Môžu existovať dokumenty s rovnakým názvom, ale musia byť uložené v rôznych repozitároch. K repozitáru môže byť pridelených viacero variantov.

Entita *ImportHistory* reprezentuje informácie o importovaní repozitárov s dokumentmi a k nim prislúchajúcich variantoch. Entita zatiaľ obsahuje atribúty o počte výnimiek, ktoré vznikli pri importovaní a o dátume, kedy bol záznam o importe vytvorený a uložený. Táto entita sa bude v budúcnosti rozširovať a pravdepodobne pribudne prepojenie na repozitár.

2.2.4 User

Pre identifikáciu používateľov sa používa entita *User*. Používateľ je identifikovaný pomocou atribútu login. Žiadny dvaja používatelia nemajú tento atribút rovnaký. Pre prihlasovanie k účtu slúži heslo,

ktoré je v systéme zašifrované. K používateľovi sú priradené viaceré dokumenty, dokument však patrí len k jednému používateľovi.

2.2.5 Annotation

K dokumentu je možné priradovať anotácie. Reprezentované sú pomocou entity *Annotation*. Anotácia má svoj typ a autora. Autor je reprezentovaný entitou *Anotator*. Táto entita vznikla zrušením vzťahu používateľa ako autora anotácie, pretože autorom nemusí byť vždy používateľ systému. Anotátor môže byť autorom viacerých anotácií, pričom anotácia má vždy len jedného autora.

Text anotácie je uložený v jej atribúte *content*. Prepojenie anotácie na dokument sa vykonáva pomocou entity *AnnotationBinding*. Určuje, kde v dokumente sa anotácia nachádza. Na tento účel slúžia atribúty *before_selected* a *after_selected* vymedzujúce okolie označeného textu.

2.2.6 Variant

Variant predstavuje verziu vzťahov v systéme a jednotlivých RDT. Variant sa viaže aj na repozitár. Jeden repozitár môže mať viac variantov a jeden variant môže patriť viacerým repozitárom. Variant obsahuje teda záznamy RDT a ich vzájomné prepojenia. Reprezentuje množinu respektíve set (verziu) týchto entít. Variant je jednoznačne určený aj atribútom *name*, ktorý nemôže byť zhodný s iným variantom.



Obr. 5. Dátový model systému.

2.3 Architektúra systému

Systém je postavený na architektúre klient – server.

Serverová časť aplikácie

- ▲ komunikuje s databázovou vrstvou
- ▲ vykonáva biznis logiku
- ▲ vytvára prezentačnú vrstvu

K serverovej časti aplikácie prístupujú tiež portály, pre ktoré systém poskytuje obsah a sú tiež v pozícii klientov. Aplikácia prijíma ich požiadavky zväčša v JSON formáte. Tento typ klientov, je nie je okrem predpísaného tvaru požiadaviek inak závislý na našom systéme.

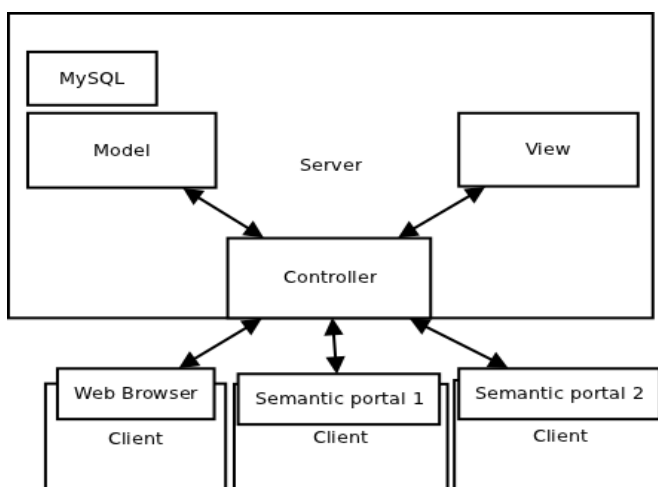
Klientská časť aplikácie je vďaka klientskym skriptom interaktívna a odľahčuje server od niektorých činností (vyhľadávanie a zoradovanie dát). Jej súčasťou je tiež pokročilé zobrazenie dát vo forme grafu ako aj práca s grafom. Klientske skripty sú tiež nápomocné pri zlepšovaní používateľského pohodlia a zefektívnení rozloženia a správania ovládacích prvkov.

Architektúra aplikácie na serveri je daná použitím aplikačného rámca Ruby on Rails, ktorý využíva architektonický vzor *Model – View – Controller*.

Model reprezentuje entity dátového modelu. Je tiež rozhraním medzi databázou a samotnou aplikáciou. Okrem jednoduchého načítania dát z databázy je doplnený o metódy, ktoré tieto dáta spracovávajú prípadne vytvárajú.

Controllery spracovávajú požiadavky klienta a riadia tok aplikácie. Zabezpečujú tiež autorizáciu používateľov. Na základe parametrov, ktoré sú dané požiadavkou klienta alebo inými vlastnosťami systému, určujú spúšťanie metód modelov. Vytvárajú tiež prezentačnú vrstvu pomocou šablón, do ktorých prípadne vkladajú dáta. Controllery sú odľahčené o aplikačnú logiku a požadovaná funkcionálna v ich akciách je vykonávaná modelmi.

View je prezentačná vrstva, tvoria ju šablóny, väčšinou v jazyku HTML, do ktorých sú vložené premenné s dátami z controllera alebo modelu. Systém tiež vytvára pohľady s dátami vo formáte JSON.



Obr. 6. Architektúra systému.

2.4 Systémové komponenty

Logicky je možné systém rozdeliť do troch skupín, a to dokumenty, anotácie a metadáta. Každá z týchto skupín obsahuje funkcionality, ktorá pracuje s danými entitami. Vyčleniť možno prihlasovanie používateľov a komunikáciu systému s jeho klientmi.

2.4.1 Prihlásenie používateľa

Pred začatím práce sa musí používateľ prihlásiť do systému. Slúži na to úvodná obrazovka kde vloží svoje meno a heslo. Tieto údaje sú totožné s údajmi v akademickom informačnom systéme. Prihlasuje sa teda pomocou IS účtu a zostáva prihlásený pokiaľ nevypne webový prehliadač alebo sa explicitne neodhlási pomocou tlačidla „Sign out“. V systéme je zatiaľ definovaná len jedna rola, ktorá má prístup ku všetkým funkciám.

Implementácia

Používateľa v systéme reprezentuje model „User“. Samotná funkcionality prihlasovania je implementovaná pomocou gemu „Devise“¹. Informácie o používateľovi uchováva v tabuľke „Users“, ktorá obsahuje atribúty modelu. Z hľadiska prihlasovania je dôležitý atribút „login“, ktorý jednoznačne identifikuje používateľa.

Pre správne fungovanie gemu devise je nutné používať konfiguračný súbor. Jeho umiestnenie je nasledovné: `config/initializers/devise.rb`.

Pohľady používané týmto gemom je možné meniť a nachádzajú sa v adresári: `app/views/devise`. Aby boli gemom používané, je nutné, aby v konfiguračnom súbore bola povolená direktíva „`config.scoped_views = true`“.

Ďalšie konfigurácie sa vykonávajú priamo v modeli pomocou tokenov. Jedným z nich je „`database_authenticatable`“, ktorý zapína prihlasovanie pomocou databázy. V prípade Come²t je však nahradený tokenom „`ldap_authenticatable`“ z dôvodu prihlasovania sa pomocou externého autentifikačného servera. Je ním univerzitný server využívajúci technológiu LDAP². Prihlasovanie týmto spôsobom implementujeme rozšírením gemu Devise pluginom „`devise_ldap_authenticatable`“³ a „`net_ldap`“.

Zmena na prihlasovanie modulom LDAP je zabezpečená vložení spomínaného tokenu „`ldap_authenticatable`“ do modelu používateľa. Výsledkom je, že prihlasovanie používateľa sa nevykoná lokálne pomocou databázy, ale pomocou autentifikačného servera. Pre správne fungovanie modulu je potrebné nastaviť konfiguračný súbor pre prístup na server. Nachádza sa v adresári: `config/ldap.yml`. Nastavenia musia byť nasledovné:

- host: ldap.stuba.sk
- port: 389
- attribute: uid
- base: ou=People,dc=stuba,dc=sk

¹ <https://github.com/plataformatec/devise>

² <http://tools.ietf.org/html/rfc4510>

³ https://github.com/dpmcnevin/devise_ldap_authenticatable

Po úspešnom prihlásení sa používateľove údaje ako login a zašifrované heslo skopírujú do lokálnej databázy.

Atribút „login“ používateľa sa používa aj v spojení s anotáciami. Pri vytvorení novej anotácie sa uloží aj jej autor. Autorom však nemusí byť len používateľ systému, ale aj používateľ externého systému, ktorý využíva služby poskytované systémom Come²t. Z tohto dôvodu bol vytvorený aj model „Anotator“. V prípade, že autorom anotácie je používateľ systému Come²t, meno používateľa sa uloží do tabuľky anotátorov v prípade, že sa v nej ešte nenachádza. Následne sa vytvorí prepojenie anotácie na jej autora. V opačnom prípade sa meno získa z externého systému.

Na odhlásenie používateľa je vytvorené menu. Je vložené do šablóny aplikácie a nachádza sa v pravom hornom rohu. Používateľovi umožňuje kedykoľvek sa odhlásiť. Zobrazuje aj jeho prihlasovacie meno.

Návrh na vylepšenie

Pre budúci vývoj systému sú navrhované nasledovné vylepšenia:

- Logovanie používateľových akcií – zaznamenávanie akcií vykonaných používateľom. Táto funkcia pomáha v sledovaní činností používateľov a ich vyhľadani napríklad v prípade poškodenia údajov v systéme.
- Pridanie rolí – definovanie viacerých rolí v systéme. Dôvodom je zabránenie prihlásenia sa pomocou ľubovoľného účtu z informačného systému univerzity. Zabráni sa tak úmyselnému poškodeniu dát v systéme.

2.4.2 Komunikácia s klientmi

Come²t spravuje obsah a metadáta pre ďalšie systémy, ktoré k jeho službám pristupujú ako klienti.

Možné scenáre komunikácie Come²t s klientmi:

C1: Pri tvorbe nového setupu si ALEF vyžiada všetky repozitáre, ktoré spravuje Come²t.

Come²t posielá zoznam repozitárov s priradenými metadátami (variantmi) v JSON formáte (Obr. 7).

URL: `cometURL/repositories/list`

C2a: Klient si vyžiada od Come²t poslednú verziu metadát vo vzťahu k obsahu.

Come²t vracia číslo verzie v JSON formáte (Obr. 8).

URL: `cometURL/repository_metadata/%ID_VARIANT%/version`

C2b: Klient si vyžiada od Come²t poslednú verziu repozitára.

Come²t vracia číslo verzie v JSON formáte (Obr. 8).

URL: `cometURL/repository/%ID_REPO%/version?key=%APIKEY%`

```
{
  repository:[
    {
      name:'FLP-Lisp',
      id:'b44e4e02d8b90',
      version:2
      metadata:[
        {
          name:'Lisp-Manual',
          id:'ac0ba4d855199',
          version:314
        },
        {
          name:'Lisp-Automatic',
          id:'44e4e02d8b908',
          version:21
        },
        ...
      ]
    },
    ...
  ]
}
```

Obr. 7. Zoznam repozitárov v JSON formáte.

```
version: 1
```

Obr. 8. Formát odpovede s verziou repozitára/metadát (C2a, C2b).

C3a: Klient si vyžiada metadáta pre svoj setup.

Come²t vracia metadáta so štruktúrou špecifikovanou pre export.

URL: cometURL/repository_metadata/%ID_VARIANT%

C3b: Come²t obsahuje novú verziu obsahu pre setup.

Come²t posielala obsah, ktorým sa nahradí obsah v danom setupe u klienta. Formát súborov je daný špecifikáciou ZIP archívu pre export repozitára. Okrem obsahu je súčasťou archívu aj XML súbor obsahujúci anotácie (Obr. 9).

URL: cometURL/repositories/%ID_REPO%/fetch?key=%API_KEY%

```

<?xml version="1.0" encoding="UTF-8"?>
  <relations xmlns="http://fiit.stuba.sk/ns/alef" type="annotation-to-lo">
    <document id="flp-book-p-0">
      <annotation id="2143" type="HighlightAnnotation">
        <selected_text>stručne</selected_text>
        <before_selected>a o logickom programovaní. V úvode </before_selected>
        <after_selected>predstavíme niektoré často používané</after_selected>
        <access>public</access>
        <state>deleted</state>
      </annotation>
      ...
    </document>
    ...
  </relations>

```

Obr. 9. Anotácie prislúchajúce k dokumentom - formát XML.

C4: Požiadavka na zmenu obsahu a metadát vyvolaná z klienta.

Odoslanie aktuálnych anotácií v JSON formáte Come²tu (Obr. 10) a presmerovanie na príslušný dokument v Come²te, kde môže byť upravovaný.

URL: cometURL/documents/update_annotations?apikey=%API_KEY%

```

[
  {
    id : 'flp-book-p-0',
    annotations: [
      {
        'id':          2143,
        'type':        'HighlightAnnotation',
        'fragment':    null,
        'selected_text': 'stručne',
        'before_selected': 'a o logickom programovaní. V úvode ',
        'after_selected': 'predstavíme niektoré často používané',
        'content':     null,
        'author':      'Jožko Púčik',
        'access':      'public',
        'state':       'new'
      },
      ...
    ]
  }
]

```

Obr. 10. Zoznam anotácií prislúchajúcich k dokumentu.

Publikovanie zmien je zabezpečené číslami verzií pre repozitáre, varianty metadát a ich väzby. Číslo verzie je možné zvýšiť, ak boli vykonané zmeny obsahu alebo metadát.

Funkcionalita zaznamenania zmien je implementovaná pomocou callback funkcií modelov „*Rdt*“, „*RdtsRelationship*“ a „*RdtDocumentRelationship*“. Vyvolané sú po uložení modelu a zmenia príznak „*has_changed*“ buď priamo v modeli, v ktorom boli vyvolané, alebo v modeli nadradenej entity (napr. „*Rdt*“ zmení „*Variant*“).

- ✦ metadáta:
 - pri zmene RDT (create, delete, edit)
 - pri zmene RDT-RDT vzťahu (create, delete, edit)
- ✦ repozitár:
 - zmena dokumentu (create, delete, edit)
 - zmena anotácie vzťahujúcej sa na dokument v repozitári
- ✦ väzba repozitár - metadáta:
 - pri zmene Dokument-RDT vzťahu (create, delete, edit)
 - zmena repozitára – „*has_changed*“ sa upravuje vo všetkých väzbách tohto repozitára

Funkcia „*publish*“ je vyvolaná z controllera na danom modeli („*Variant*“, „*Repository*“, „*RepositoryVariant*“), čím sa zmení atribút „*version*“ a príznak „*has_changed*“ je opäť nastavený na false. Pri zverejnení zmien repozitára sa tiež kontroluje prítomnosť dokumentov, na ktoré sa vzťahujú anotácie s konfliktmi. Takéto dokumenty majú nastavený príznak „*has_conflicts*“.

Pre komunikáciu ohľadne obsahu je možné používať tajné API kľúče, ktoré sa generujú pri vytvorení repozitára.

2.4.3 Dokumenty

Atribúty entity Document:

id	Unikátne ID
content	Obsah dokumentu
created_at	Dátum a čas vytvorenia
updated_at	Dátum a čas poslednej úpravy
deleted	Príznak vymazania dokumentu
repository_id	ID repozitára
name	Názov dokumentu
version	Číslo verzie
user_id	ID používateľa
previous_version	Číslo predchádzajúcej verzie
is_valid	Príznak validity dokumentu
has_conflicts	Príznak konfliktných anotácií
document_type_id	ID typu dokumentu
alef_id	Textový identifikátor používaný napr. v klientkom systéme
html_content	Uložený transformovaný obsah dokumentu

Vytvorenie a úprava dokumentu

XML editor vo formulári pre vytváranie a úpravu dokumentov vznikol rozšírením štandardného textového poľa HTML formuláru a jeho obsah je bez akýchkoľvek úprav ukladaný ako atribút „*content*“.

Import súboru do editora je zabezpečený asynchrónnou požiadavkou. Ruby gem „*remotipart*“ umožňuje nahranie súborov na server prostredníctvom technológie AJAX. Nahraný súbor je na serveri prečítaný a jeho obsah je vložený do skriptu, ktorý naplní textové pole vo formulári. Skript je vložený do IFRAME elementu vytvoreného gemom „*remotipart*“ a následne je vykonaný. Pri vkladaní obsahu vybraného súboru do textového poľa je potrebné znovu vytvoriť XML editor aby sa vložený obsah zobrazil so zvýraznenou syntaxou.

Vymazanie dokumentu je možné iba zmenou atribútu „*deleted*“ na hodnotu true. Takto vymazaný dokument sa potom nikde v systéme nezobrazuje ale je ho možné obnoviť a opäť s ním pracovať. Obnovenie zabezpečuje akcia „*undestroy*“ v DocumentsControlleri.

Repozitáre

Generovanie API kľúča

Vykonávané v:

- model repozitára (*app/models/repository.rb*)

Použité funkcie:

- *generateApiKey*

Popis:

Používa kryptografickú hašovaciú funkciu SHA-1 pre vygenerovanie 32 miestneho reťazcu na základe aktuálneho času. Ten slúži ako API kľúč pri pridávaní nového repozitára.

Presúvanie dokumentov

Feature je začlenená do obrazovky RD1, čiže detail repozitára. Používateľ si môže označiť jeden alebo viacero dokumentov, zvoliť jeden z existujúcich repozitárov a presunúť zvolené dokumenty z aktuálneho repozitára do zvoleného repozitára. Feature pracuje s entitami *Document* a *Repository* a s atribútom *repository_id*. Po presunutí dokumentov sa zobrazí detail aktuálneho repozitára, z ktorého boli dokumenty presunuté. K presunutým dokumentom môže používateľ pristupovať z jeho nového umiestnenia.

Implementácia

Implementácia bola prevedená v controlleri *documents_controller* a vo view *repositories_documents*. Implementovaná bola metóda *move*, ktorá napĺňala samotnú funkcionality feature zmenou ID rodičovského repozitára vybraných dokumentov. Okrem toho bola doplnená možnosť výberu jedného a viacerých dokumentov v detaile repozitára a možnosť výberu cieľového repozitára z existujúcich, takisto v obrazovke RD1.

Súčasný stav

V súčasnosti je možné zvoliť si v repozitári jeden alebo viac dokumentov zaškrknutím checkboxov, ktoré sa pri nich nachádzajú. Ak nie je vybraný žiaden dokument, tak sa po kliknutí na tlačidlo *move* zobrazí hlásenie, vyzývajúce k výberu dokumentu.

Po vybratí zvolených dokumentov a kliknutí na tlačidlo *move* sa otvorí okno so zobrazenými repozitármi, do ktorých je možné dokumenty presunúť. Po zvolení cieľového repozitára a kliknutí na tlačidlo *move* sa dokumenty presunú do cieľového repozitára, ktorý sa zobrazí namiesto zdrojového.

Návrh na vylepšenie

V budúcnosti je možné zabrániť používateľovi manipulovať s obsahom repozitára po otvorení okna pre presúvanie dokumentov. V súčasnosti je možné označovať a odznačovať dokumenty aj po kliknutí na tlačidlo *move*. Rovnako je potrebné ošetriť prípad, kedy používateľ nezvolil žiaden cieľový repozitár.

Import repozitára a metadát

Funkcia importu slúži na vkladanie veľkého množstva dát do systému. Dáta sú reprezentované archívom, ktorý si používateľ zvolí. Pred importom je používateľovi umožnený výber položiek, ktoré sa v archíve nachádzajú, v prípade, že nepotrebuje importovať celý jeho obsah. Na výber má aj zapnutie validácie importovaných dokumentov a metadát. Znázornenie tohto dialógu je na Obr. 11. Po potvrdení voľby sa spustí akcia importu a po jej skončení sa zobrazí informácia o ukončení akcie.

Pri importovaní sa evidujú aj konflikty a ich počet sa ukladá do záznamu v databáze. Počet konfliktov tvorí základ pre model histórie importov, ktorý je popísaný v časti „*História importu*“.

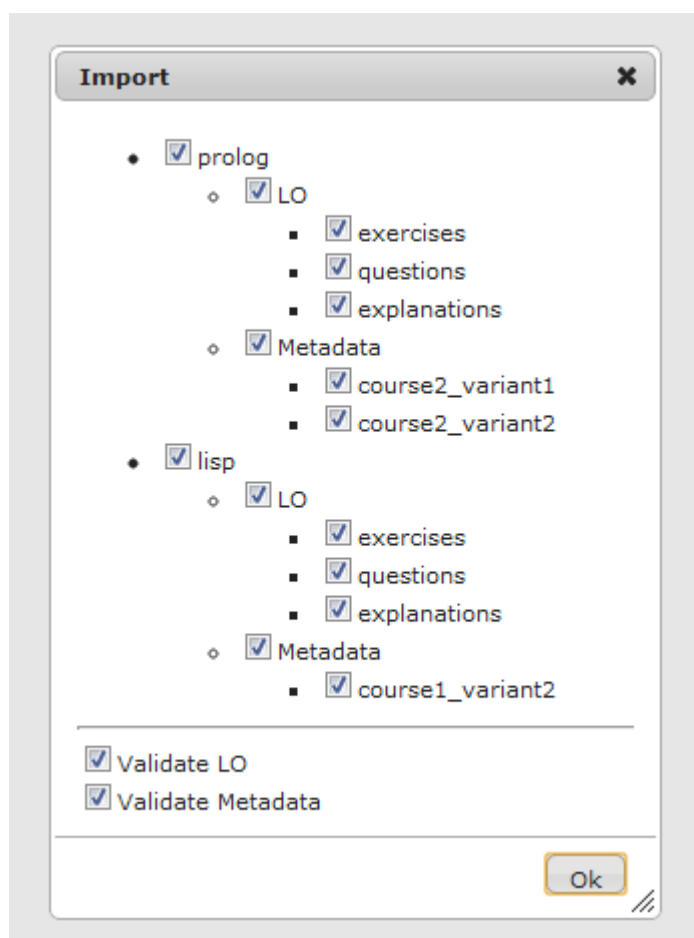
Implementácia

Štruktúra importovaného archívu je daná a musí byť dodržaná. Formát archívu je „...zip“. Archív obsahuje jeden alebo viac kurzov. Každý kurz predstavuje jeden adresár v archíve. Obsahuje adresár pre každý typ dokumentu (exercises, explanations, questions). Ďalej obsahuje adresár „resources“, kde sa nachádzajú obrázky k dokumentom. Štruktúra adresára „metada“ obsahuje pre každý variant jeden alebo viac adresárov (Variant1, Variant2 ...). Názov variantu sa bude čerpať z názvu adresára, v ktorom sa metadáta nachádzajú. Štruktúru archívu možno vidieť na Obr. 12.

Výber archívu na importovanie sa uskutočňuje kliknutím na tlačidlo „Import“ v obrazovke repozitárov, ktoré zobrazí formulár na výber súboru. Po potvrdení výberu sa archív skopíruje do systému do adresára s názvom podľa identifikačného čísla používateľa. Týmto sa zaisťuje, že v systéme môže prebiehať viac akcií importovania naraz bez toho, aby si používatelia zasiahli do súborov. Následne sa analyzuje obsah archívu metódou „items“. Vytvorí sa JSON objekt, ktorý sa pošle prezentačnej vrstve. Tá ho interpretuje pomocou dialógu na výber možností pri importovaní – Obr. 11. Jednotlivé možnosti a výber súborov sa vykonáva odškrtnutím checkboxov v dialógu. Usporiadanie elementov v dialógovom okne pripomína stromovú štruktúru, aby bola lepšie čitateľná.

Po potvrdení voľby sa odošlú vybrané položky formou hashu. Importovanie prebieha asynchrónne. Zaisťuje ho spustenie POST akcie v samostatnom procese. Najprv sa importujú všetky dokumenty nazývané „learning object“ (LO). Vkladajú sa do repozitára, ktorého meno je odvodené od názvu kurzu. Pri importovaní dokumentu sa nájdu všetky k nemu prislúchajúce obrázky. Uložia sa v systéme formou „assetov“. Takýmto uložením získajú iný názov, ktorý nekorešponduje s názvom v dokumente. Pri vytváraní náhľadu by sa preto nezobrazovali obrázky, obsiahnuté v dokumente.

Z tohto dôvodu sa po uložení obrázkov do systému prepisuje obsah dokumentu novými názvami obrázkov.

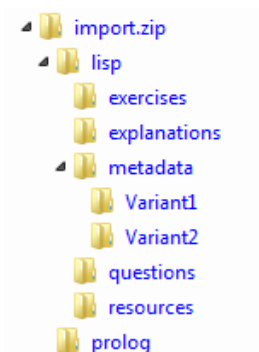


Obr. 11. Možnosti importovania.

Po dokumentoch sa importujú metadáta, ktoré vytvoria variant. Ak daný variant už existuje, nevytvára sa nový. Pri nájdení duplikácie sa daný objekt preskočí a pokračuje sa ďalšou položkou. V takomto prípade sa inkrementuje počet konfliktov v zázname o histórii importov. Po importovaní metadát sa vytvoria medzi nimi vzťahy.

Ak používateľ zapol validácie importovaných súborov, vykonajú sa pre každú položku archívu. Dokumenty sa overujú podľa šablóny rovnakým spôsobom ako pred vytvorením ich náhľadu. Vzťahy medzi metadátami sa overujú podľa šablón umiestených v adresári: *public/*.

Po ukončení importu sa zmažú dočasné súbory v adresári s menom používateľovho identifikačného čísla. Odošle sa odpoveď na POST request vyvolávajúci akciu importu, ktorá zobrazí dialóg o ukončení akcie.



Obr. 12. Štruktúra archívu.

Návrh na vylepšenie

Pre vylepšenie funkcionality importu sú navrhované nasledovné vylepšenia:

- Zmena spôsobu vytvorenia asynchrónneho vykonania akcie importu obetovaním procesu systému.
- Lepšie zobrazenie a výber možností importu.
- Zobrazenie sumáru po vykonaní importu.
- Hláška indikujúca úspech akcie sa nezobrazí v prípade opustenia obrazovky počas vykonávania importovania aj napriek tomu, že úspešne skončí. Dôvodom je zlyhanie POST requestu pri zmene obrazovky. Je vhodné zmeniť spôsob jej spustenia.

História importu

Systém má vytvorený experimentálny model pre históriu importovania dokumentov. Obsahuje atribúty určujúce počet konfliktov, ktoré vznikli pri importovaní a čas začiatku a konca importovania.

Implementácia

História importovania je vytvorená pomocou modelu „*import_history*“. Obsahuje tri atribúty, keďže ide o experimentálnu funkcionality. Medzi ne patrí:

- `num_errors` – počet konfliktov, ktoré vznikli pri importovaní
- `created_at` – čas spustenia importovania archívu
- `updated_at` – čas ukončenia importovania archívu

Zaznamenávanie konfliktov sa vykonáva pomocou metódy „*increment_errors*“, ktorá zväčšuje číselnú hodnotu uloženú v atribúte „*num_errors*“. Nový záznam sa vytvorí vždy pri spustení importu archívu. Konflikt pri importovaní vzniká, ak sa pri tomto procese v systéme nájde entita, zodpovedajúca práve importovanej.

Zobrazenie histórie nie je zatiaľ priamo prístupné zo systému. Dá sa zobraziť pomocou zadania url „*import_histories/:id*“, kde „*:id*“ reprezentuje identifikačné číslo záznamu v databázovej tabuľke histórií.

Návrh na vylepšenie

Táto časť systému má veľký priestor na vylepšovanie. Je potrebné rozšíriť počet atribútov v zázname o importovaní. Chýbajú typy konfliktov a ich prehľadné zobrazenie. Takisto je nutné zlepšiť pohľady zobrazujúce jednotlivé záznamy a vytvoriť v systéme tlačidlo, pomocou ktorého sa pohľady dajú

vyvolať – zatiaľ nie sú v systéme priamo prístupné a dajú sa zobrazíť len pomocou zadania požadovanej url.

Export repozitára

Export repozitára je jeho prevedenie na štruktúrovaný ZIP archív.

Obsah dokumentov je uložený v samostatných XML súboroch, rozdelených v adresároch podľa typu dokumentu.

Metadáta sú rozdelené do adresárov podľa variantov. Zoznam RDT je uložený v súbore *concepts.xml*. Vzťahy sú podľa typov rozdelené v samostatných súboroch.

K dokumentom môžu byť priradené obrázky, ktoré sú v archíve uložené v adresári *resources*.

Export používa metódy modelov „*Repository*“ a „*Variant*“.

Vytvorenie archívu a spustenie všetkých potrebných funkcií vykonáva metóda export modelu „*Repository*“.

- *create_concepts_xml* – vytvorenie XML súboru so zoznamom RDT pre daný variant
- *create_relationships_xml* – XML súbor so zoznamom vzťahov RDT-RDT a Dokument-RDT
- *create_annotation_xml* – XML súbor so zoznamom anotácií priradených dokumentom z daného repozitára
- *create_documents_xml* – vytvorí XML súbory s obsahom dokumentov v danom repozitári.
- *create_asset_files* – vloženie súborov priradených dokumentom z daného repozitára do ZIP archívu

Exportovať je možné viacero repozitárov naraz. Pred samotným exportom sú nájdené nevalidné a konfliktné súbory. Nevalidné súbory nie sú exportované.

Validácia dokumentu

Validácia obsahu dokumentu sa spúšťa na požiadanie používateľa pri ukladaní dokumentu. Obsah je reprezentovaný vo forme XML. Úlohou validácie je zobrazenie vzniknutých chýb a zachovanie správnosti dokumentu. Dokument je možné uložiť aj bez spustenia validácie.

Implementácia

Obsah dokumentu je v systéme Come²t reprezentovaný pomocou XML vo formáte DocBook⁴. Jeho správnosť sa overuje pomocou schémy definovanej v jazyku RELAX NG⁵. Nachádza sa v adresári: `public\docbook-5.0-alef-1.0\docbook-alef.rnc`. Predstavuje rozšírenie štandardnej schémy DocBook-u o elementy špecifické pre systém ALEF⁶.

Validácia obsahu sa zapína pomocou checkboxu umiestnenom pod obsahom dokumentu. Ak je zaškrtnutý, pred uložením dokumentu sa zavolá metóda „*validate*“ definovaná v modeli dokumentu. Overenie správnosti vykoná pomocou gemu s názvom „*Nokogiri*“⁷. Jedná sa o wrapper pre knižnicu „*libxml*“⁸. Na vstupe metódy sa nachádza obsah dokumentu a schéma, ktorá sa má použiť pri validácii. Schéma sa pred použitím spracuje na inštanciu triedy *Nokogiri::XML::RelaxNG*. Výstupom metódy je pole chýb nájdených v obsahu.

Ak je pole chýb prázdne, dokument sa úspešne uloží. V opačnom prípade sa uskutoční výpis všetkých jeho položiek. Príklad takéhoto výpisu je na Obr. 13. Používateľovi sa v ňom zobrazia nasledovné atribúty:

- code - kód chyby
- level – level atribútu
- line – riadok, v ktorom sa chyba nachádza
- message – text chyby
- str1 – názov elementu



Obr. 13. Výpis chýb obsahu dokumentu.

Návrh na vylepšenie:

Z dôvodu lepšej adaptivity pre klientský portál je možné implementovať výber šablón pre validáciu dokumentu. Používateľ by mal na výber viacero šablón pred spustením validácie zo zoznamu. V zozname by bola vždy vopred označená východzia šablóna.

Kontrola integrity dokumentu

Súbory priložené k dokumentu sú linkované pomocou XML elementov `<imagedata>`. Kontroluje sa preto, či súhlasia `imagedata` elementy v dokumente so súbormi, ktoré používateľ k súboru priložil

⁴ <http://www.docbook.org/>

⁵ <http://relaxng.org/>

⁶ Slúži ako vzor adaptívneho portálu

⁷ <http://nokogiri.org/>

⁸ <http://libxml.rubyforge.org/rdoc/>

(prípadne boli vložené pri importe repozitára).

Kontrola prebehne iba v prípade, že ju používateľ povolí v rozhraní aplikácie. Tak je umožnené priebežné ukladanie dokumentov, ktoré napríklad zatiaľ nemajú pripravené obrázkové prílohy.

Podmienka v akcii „*update*“ v „*DocumentsController*“ – zistenie, či bolo označené zaškrtnuté pole vo formulári dokumentu.

Z akcie controlleru sa volá funkcia „*check_integrity*“ modelu „*Document*“.

Overenie obrázkových príloh je zabezpečené nájdením imagedata elementov pomocou Xpath. Z atribútu fileref sa získava názov súboru, ktorý sa následne použije ako parameter SQL dopytu.

V databáze sa vyhľadá záznam o súbore s daným názvom a referenciou na kontrolovaný dokument. Ak sa takýto súbor nenájde, príznak „*image_conflict*“ nadobudne hodnotu true a do poľa errors sú pridané informácie o chýbajúcom súbore.

View a preview dokumentu

Pri práci s dokumentmi je potrebné vidieť, ako bude dokument vyzeráť, keď sa zobrazí vo webovom portáli. Túto funkcionality umožňujú náhľad na dokument, ktorý ho zobrazí vo formáte HTML. Celkovo rozlišujeme tri typy náhľadov:

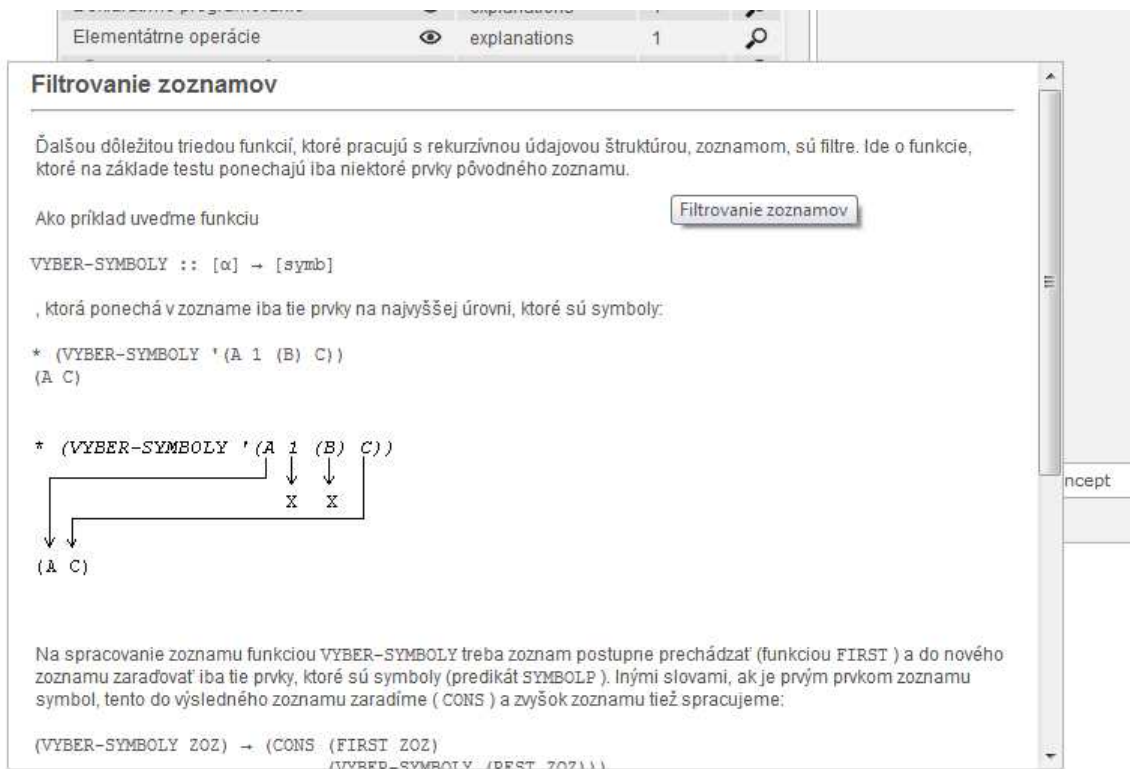
- View – zobrazenie uloženého obsahu dokumentu vo formáte HTML presmerovaním na príslušnú obrazovku. Uskutočňuje sa tlačidlom „*View*“ v obrazovke editácie dokumentu.
- Preview – zobrazenie práve editovaného obsahu dokumentu vo formáte HTML pod editovacím poľom. Spúšťa sa tlačidlom „*Preview*“ pod poľom editácie obsahu dokumentu.
- Rýchly preview v obrazovke RDV1 - zobrazenie uloženého obsahu dokumentu vo formáte HTML vo forme tooltipu. Tento náhľad sa zobrazí ponechaním kurzora nad ikonou „oka“. Ikona sa nachádza pri názve dokumentu v obrazovke RDV1. Náhľad sa v tom prípade uskutoční asynchrónnym volaním metódy na transformáciu obsahu dokumentu. Zobrazí sa formou tooltipu, čo možno vidieť na Obr. 14.

Implementácia

Keďže obsah dokumentu je reprezentovaný pomocou XML, je možné pretransformovať ho do tvaru v jazyku HTML. Na tento účel slúžia transformácie v jazyku XSLT. K dispozícii je transformácia v adresári: *public\docbook-xsl-alef-1.0\alef2xhtml.xsl*.

Obsah dokumentu vo formáte HTML sa zobrazí len v prípade, ak je dokument validný. Táto skutočnosť sa overí pred zobrazením náhľadu.

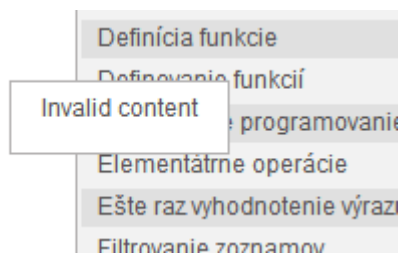
2 Opis prototypu



Obr. 14. Náhľad formou tooltipu.

Ak je obsah dokumentu nevalidný, namiesto náhľadu sa zobrazí upozornenie pre používateľa. Príklad zobrazenia chyby v náhľade je na Obr. 15, ktorý obsahuje náhľad z obrazovky RDV1.

Model dokumentu uskutočňuje transformáciu metódou „*trans_to_html*“, ktorá používa gem *Nokogiri*. Vstupom je transformačný súbor. Výstupom je obsah dokumentu vo formáte HTML.



Obr. 15. Chybová hláška pri zobrazení nevalidného obsahu.

Návrh na vylepšenie

Funkcionalitu systému možno rozšíriť umožnením výberu vlastnej transformácie používateľom. Tento výber by prebiehal pred spustením náhľadu zo zoznamu. V zozname by bola vždy vopred označená východzia transformácia.

Verziovanie

Dokumenty sú verziované, čiže používateľ má po uložení zmien v dokumente možnosť vrátiť sa k predchádzajúcej verzii dokumentu.

Verzie sa v prípade vymazania dokumentu nestratia, to znamená, že dokument zostáva v databáze, ale je ukrytý. Akciou „*undestroy*“ v „*DocumentsControlleri*“ môže byť opäť vrátený do pôvodného – viditeľného stavu.

Model „*Document*“ je rozšírený gemom „*Acts_as_versioned*“, ktorý zabezpečuje automatické vytváranie verzií pri akciách vytvorenia a úpravy dokumentu. Verzia dokumentu pritom uchováva atribúty „*content*“, „*name*“, „*user_id*“, „*has_conflicts*“ a „*html_content*“.

Čísla verzií sú pridelované v postupnosti, v akej verzii vznikali. Dokument je možné vrátiť do ktorejkoľvek predchádzajúcej verzie, čím sa tiež zmení atribút „*version*“. Úpravou a uložením takéhoto dokumentu vzniká nová verzia.

Atribút „*previous_version*“ uchováva číslo predchádzajúcej verzie dokumentu. Tento atribút umožňuje vytvoriť strom verzií a sledovať históriu dokumentu.

Pre každú verziu je možné zobrazíť zmeny obsahu oproti predchádzajúcej verzii. Používa sa pri tom funkcia ruby gemu *Diffy*. Zobrazujú sa potom zmenené riadky s vyznačenými úsekmi, ktoré boli upravené alebo zmazané/pridané.

Vytvorením novej verzie sa skopírujú všetky anotácie, ktoré boli k dokumentu priradené.

Nie je možné uložiť nezmenený dokument. Overenie, či nastali zmeny, je zabezpečené funkciou gemu *Acts_as_versioned version_condition_met*, ktorá bola rozšírená o ďalšie atribúty, ktorých zmena nespôsobuje zmenu verzie celého dokumentu.

Vzťahy medzi RDT a dokumentmi

Používateľ môže vytvárať, upravovať a odstraňovať vzťahy medzi dokumentmi a pojmami. Používateľ môže vytvárať, upravovať a odstraňovať typy vzťahov medzi pojmami a dokumentmi. Vytvorená bola entita *RDTDocumentRelationshipType* s atribútom názov. Ďalej bola vytvorená entita *RDTDocumentRelationship*, s atribútmi váha, ID RDT a ID dokumentu, medzi ktorými je vzťah. Bola prepojená na entity *RDT*, *Document* a *RDTDocumentRelationshipType*. Po vytvorení typu vzťahu alebo vzťahu bude typ vzťahu alebo vzťah uložený v databáze a používateľ bude mať k nemu prístup cez používateľské rozhranie. Pri vytváraní vzťahu sa vo formulári v poli „*Type*“ zobrazia na výber existujúce typy vzťahov.

Implementácia

Implementované boli modely s názvom *RDTDocumentRelationshipType* a *RDTDocumentRelationship*. Model *RDTDocumentRelationshipType* má povinný atribút *name*. Model *RDTDocumentRelationship* má povinné atribúty *weight* (váha), *from_id* (ID RDT) a *to_id* (ID dokumentu). V aplikačnej časti sú implementované funkcie na vytvorenie, zmazanie, upravenie typu vzťahu medzi dokumentom a RDT a obdobné funkcie pre prácu so vzťahom medzi dokumentom a RDT.

Súčasný stav

V súčasnosti je možné pridávať a odstraňovať vzťahy medzi dokumentmi a RDT v obrazovke view dokumentu v záložke „RDT“. Pridávať vzťahy je možné kliknutím na tlačidlo „Add“, kedy sa používateľ presunie do obrazovky pre pridávanie vzťahu. Tam má možnosť nastaviť typ vzťahu, váhu, odkiaľ a kam pôjde a či bude orientovaný. Po kliknutí na „*Create RDT - Document relationship*“ sa vytvorí vzťah medzi dokumentom a RDT. V prípade, že neboli všetky hodnoty správne, vzťah sa nevytvorí a vypíše sa upozornenie.

Po vytvorení vzťahu sa používateľ presunie na obrazovku, zobrazujúce informácie o novom vzťahu. Kliknutím na tlačidlo „*Edit*“ je možné vzťah upraviť, kliknutím na tlačidlo „*Back*“ sa používateľ

presunie na obrazovku, ktorá obsahuje všetky vytvorené vzťahy. Z tejto obrazovky je možné vytvorené vzťahy upravovať, zobrazovať a mazať a takisto pridať nový vzťah.

Možné vylepšenia

V budúcnosti je možné pridať možnosť upravovania vzťahov priamo do view dokumentu. Takisto je vhodné nastaviť po editácii, vymazaní alebo vytvorení vzťahu návrat do predošlej obrazovky, teda do view dokumentu.

Vzťahy medzi dokumentmi

Používateľ môže vytvárať, zobrazovať, upravovať a mazať vzťahy medzi dokumentmi. Používateľ môže vytvárať, zobrazovať, upravovať a mazať typy vzťahov medzi dokumentmi. Je vytvorená entita *DocumentRelationship* s atribútmi *weight*, *document_type_id*, *from_document_id* a *to_document_id*, prepojená na entitu *Document* a entita *DocumentRelationshipType* s atribútom *name*, prepojená na entitu *DocumentRelationship*. Po vytvorení typu vzťahu alebo vzťahu bude typ vzťahu alebo vzťah uložený v databáze a používateľ bude mať k nemu prístup cez používateľské rozhranie. Pri vytváraní vzťahu sa vo formulári v poli „Type“ zobrazia na výber existujúce typy vzťahov.

Implementácia

Implementované boli modely s názvom *DocumentRelationship* a *DocumentRelationshipType*. Model *DocumentRelationship* má povinné atribúty *weight*, *relationship_type_id*, *from_document_id* a *to_document_id*. Model *RDTDocumentRelationshipType* má povinný atribút *name*. V aplikačnej časti sú implementované funkcie na vytvorenie, zmazanie, zobrazenie a upravenie vzťahu a typu vzťahu medzi dokumentmi.

Súčasný stav

V súčasnosti je možné pridávať a odstraňovať vzťahy medzi dokumentmi v obrazovke view dokumentu v záložke „Documents relationships“. Pridávať vzťahy je možné kliknutím na tlačidlo „Add“, kedy sa používateľ presunie do obrazovky pre pridávanie vzťahu. Tam má možnosť nastaviť typ vzťahu, váhu, odkiaľ a kam pôjde a či bude orientovaný. Po kliknutí na „Create Document relationship“ sa vytvorí vzťah medzi dvomi dokumentmi. V prípade, že neboli všetky hodnoty správne, vzťah sa nevytvorí a vypíše sa upozornenie.

Po vytvorení vzťahu sa používateľ presunie na obrazovku, zobrazujúcu informácie o novom vzťahu. Kliknutím na tlačidlo „Edit“ je možné vzťah upraviť, kliknutím na tlačidlo „Back“ sa používateľ presunie na obrazovku, ktorá obsahuje všetky vytvorené vzťahy. Z tejto obrazovky je možné vytvorené vzťahy upravovať, zobrazovať a mazať.

Možné vylepšenia

V budúcnosti je možné pridať možnosť upravovania vzťahov priamo do view dokumentu. Takisto je vhodné nastaviť po editácii, vymazaní alebo vytvorení vzťahu návrat do predošlej obrazovky, teda do view dokumentu.

2.4.4 Anotácie

Reprezentácia anotácií

Implementácia

Anotácie sú reprezentované štyrmi atribútmi:

- *selected_text* - označený text anotácie
- *before_selected* - fragment textu pred označeným textom
- *after_selected* - fragment textu po označenom texte
- *content* - samotný pridaný obsah anotácie (napr. komentár, opis chyby, ...)

Podľa atribútov *selected_text*, *before_selected* a *after_selected* a prípadne ešte podľa poradia výskytu zvoleného slova je možné určiť pozíciu slova a vyznačiť ho v texte.

Súčasný stav

Po vyznačení textu má používateľ možnosť pridať novú anotáciu. Po jej pridaní sa do atribútov *content* uloží obsah používateľa, do atribútu *selected_text* sa uloží text, ktorý používateľ vyznačil, do *before_selected* sa uložia dve predchádzajúce slová a do *after_selected* dve nasledujúce.

Anotácie a anotačný pásik

Vo view dokumentu má používateľ možnosť pridávať, upravovať a mazať anotácie. Tieto akcie môže prevádzať buď kliknutím na akcie v záložke „Annotations“ alebo kliknutím na vytvorenú anotáciu. Vytvoriť novú anotáciu môže používateľ len jej vyznačením v texte a kliknutím na zobrazené tlačidlo „New“.

Vedľa náhľadu na dokument je možné vidieť anotačný pásik, ktorý zobrazuje vytvorené anotácie. Klikaním na jednotlivé anotácie v pásiku je možné ich editovať, prípadne meniť ich polohu.

V prípade, že sa anotácie prekrývajú, je v pásiku možné zobraziť všetky prekrývajúce sa anotácie nájdením na ne kurzorom. Po nájdení kurzorom na miesto v pásiku, kde sú reprezentované prekrývajúce sa anotácie sa zobrazí dialóg, zobrazujúci zoznam týchto anotácií a akcie, ktoré je možné s nimi vykonať.

Riešenie konfliktov

Anotácia je konfliktná ak nie je možné nájsť reťazec „*selected_text*“ v obsahu dokumentu. Zároveň sa zisťuje, či zostalo nezmenené okolie anotovaného textu.

V prípade, že

- ▲ okolie aj anotovaný text zostali nezmenené, anotácia je bez konfliktov,
- ▲ okolie je zmenené a anotovaný text je nezmenený, vytvorí sa nové okolie a uloží pre väzbu anotácie na verziu dokumentu,
- ▲ okolie aj anotovaný text sú zmenené – to znamená, že hľadanie reťazca bolo neúspešné. Podľa okolia sa snaží Come²t zrekonštruovať anotovaný text. Anotácia je označená ako konfliktná a dokument má zmenený príznak „*has_conflicts*“ na true.

Kontrola konfliktov prebieha pri kopírovaní väzieb anotácií na novú verziu dokumentu po jeho úprave.

Zistenie konfliktu anotácie zabezpečuje metóda „*is_conflict*“ modelu „*AnnotationBinding*“. Hľadanie vhodného okolia vykonáva metóda „*get_unique_fragment*“ modelu „*Annotation*“.

2.4.5 Metadáta

Metadáta sú v našom systéme reprezentované ako súbor kľúčových slov - pojmov relevantných k doméne obsahu dokumentov (*RDT - Relevant Domain Term*) a súbor vzťahov medzi týmito pojmami. Dokopy tvoria jeden variant metadát. Variant sa viaže na repozitár s dokumentmi, ktorý predstavuje napr. jeden kurz vo výučbovom systéme. Na jeden repozitár môže byť naviazaných viacero variantov. Metadáta predstavujú sémantiku obsahu dokumentov a slúžia napr. na vyhľadávanie alebo odporúčanie obsahu v klientoch nášho systému.

Varianty

Variant metadát je v systéme reprezentovaný entitou „*Variant*“ a má jediný atribút - názov.

Implementácia

CRUD nad variantmi bol vytvorený pomocou Rails generátora. Metódy na úpravu a vytvorenie variantu sú realizované ako AJAX-ové volania.

Vykonávané v:

- View (*app/views/repositories/_variants.html.erb*)
- Controller (*app/controllers/variants_controller.rb*)
- Model (*app/models/variant.rb*)
- AJAX-ové volania (*app/assets/javascripts/repositories.js.coffee*)

Vzťahy medzi metadátami a repozitármi

Jednému repozitáru môže byť priradených viacero variantov metadát, aby dokumentom mohli byť priradené pojmy z variantu. Takisto jeden variant môže byť priradený viacerým repozitárom, preto je medzi nimi vzťah M:N.

Implementácia

Vzťah medzi variantom metadát a repozitárom je v systéme reprezentovaný entitou „*Repository_Variant*“. Pridávanie a odoberanie variantov metadát je realizované pomocou AJAX-ových volaní.

Vykonávané v:

- View, kde je možné priradovať varianty repozitárom (*app/views/repositories/_add_variant_to_repository.html.erb*)
- Controller (*app/controllers/repositories_controller.rb* (metóda *update*))
- Model (*app/models/repository_variant.rb*)
- AJAX-ové volania (*app/assets/javascripts/repositories.js.coffee*)

RDT a vzťahy medzi RDT

Pojem je v systéme reprezentovaný entitou „*RDT*“ a má jediný atribút - názov, vzťah medzi pojmami entitou „*RdtsRelationship*“. Vzťah je prepojený na entitu predstavujúcu typ vzťahu „*RdtsRelationshipType*“. Typy vzťahov sú v systéme „natvrdo“ nemožno ich zatiaľ meniť pomocou UI. Typ vzťahu má atribút „*oriented*“, ktorý vyjadruje, či typ vzťahu je orientovaný. Samotný vzťah má atribúty „*from_rdt_id*“, „*to_rdt_id*“, „*weight*“ (váha vzťahu z intervalu $(0,1>$), „*relationship_type_id*“.

Implementácia

Úprava, vytváranie a vymazávanie pojmov a vzťahov sa deje na klientovi a je realizované pomocou JavaScriptu. Zmeny nad týmito entitami sa pri každej operácii prejavujú v tabuľkovom zobrazení, v grafe a v štruktúre „*rdtList*“, ktorá uchováva aktuálne dáta na klientovi. Po zvolení uloženia zmien variantu metadát sa odošlú dáta z tejto štruktúry a uložia sa do databázy (opísané v ďalšej kapitole Graf - Odoslanie úprav).

Vykonávané v:

- View
 - *app/views/variants/show.html*
 - *app/views/variants/_rdts.html*
 - *app/views/variants/_rdtsrelationships.html*
- Controller (*app/controllers/variants_controller.rb*)
- Model (*app/models/variant.rb*)
- JavaScript (*app/assets/javascripts/variants.js.coffee*)

Graf

Vykonávané v:

- Zobrazenie variantu (*variants/show.html.erb*)
- javascript grafový súbor (*assets/javascripts/graph_view.js*)
- knižnica JIT (*assets/javascripts/jit.js*)

Použité technológie:

- JIT: grafová knižnica (<http://thejit.org/>)
- jQuery - dialog

Operácie súvisiace s grafom:

- Načítanie údajov
- Nastavenie grafu
- Zobrazenie grafu
- Vykonávanie úprav/vizuálnych zmien
- Odoslanie úprav
- Spracovanie úprav

Načítanie údajov

Popis:

Samotná knižnica JIT nedokáže poskytnúť funkcionality, ktoré sú potrebné pre úpravy grafu a ich následné uloženie. Údaje sú teda načítavané do dvoch separátnych premenných. Jedna z nich slúži ako vstup pre vizualizovanie grafu (premenná s názvom *json*) a v druhej sa ukladajú vykonávané zmeny tak, ako je potrebné (*rdtList*)

Použité funkcie:

- *init*
- *initCombos*

Vykonanie:

1. Vo view sa inicializujú premenné: *rdts*, *relationships*, *relationshipTypes*
2. Použijú sa ako parametre funkcií *init* a *initCombos*
3. Vo funkcii *init* sa ich hodnoty vložia do pola *rdtList*
4. Z *rdtList* sa nahrajú hodnoty do premennej *json*
5. Načíta sa graf:
 - a. ak *json* nie je prázdny (obsahuje aspoň jeden vrchol) použije sa ako vstupná premenná pre knižnicu JIT
 - b. inak je inicializácia vykonaná až pri prvom vložení uzlu do grafu
6. Vo funkcii *initCombos* sa naplnia combo boxy použité pri vytváraní nových vzťahov

Nastavenie grafu

Použité funkcie:

1. *init*
2. *\$jit.ForceDirected* - konštruktor

Vykonanie:

1. Nasledujúce akcie sú vykonané v rámci konšuktora pre novú premennú *fd*
2. Nastaví sa element, do ktorého sa vloží graf: *infovis*
3. Nastaví sa spôsob navigácie v grafe: *Navigation*
4. Nastavia sa všeobecné vlastnosti pre hrany a vrcholy: *Node*, *Edge*
5. Nastavia sa udalosti na ktoré bude graf reagovať: *Events*

Zobrazenie grafu

Použité funkcie:

- *\$jit.ForceDirected.Plot.NodeTypes.implement*
- *loadJSON*
- *computeIncremental*
- *animate*

Vykonanie:

1. pomocou funkcie *implement* poskytnutou knižnicou JIT sú vytvorené 3 typy uzlov. Každý z nich má iné farebné vlastnosti:

- a. *myNodeType* – obyčajný uzol
 - b. *mySelectedNodeType* – vybraný uzol, zvýraznený
 - c. *myNeighbourNodeType* – susedné uzly vybraného uzlu, zvýraznené
2. premennej *fd* sa nastaví vstupná premenná *json* pomocou funkciu *loadJSON*
 3. *fd* použije funkciu na výpočet pozícií uzlov, ak ju ešte nemajú zadefinovanú
 4. graf je vizualizovaný funkciou *animate*

Vykonávanie úprav/vizuálnych zmien

V grafe je možné vykonávať nasledujúce vizuálne úpravy za pomoci funkcií:

- Centrovanie uzlu
 - *centreNode*
- Ohraničenie zobrazenia od zvoleného uzlu
 - *constrainNode*
 - *constrainRecursion*
- Zvýraznenie vrcholu a jeho susedov
 - *highLightNode*
- Filtrovanie podľa vzťahov
 - *filterRelationType*
- Znovu vygenerovanie pozícií
 - *reset*

Úpravy RDT a ich vzťahov v rámci grafu sú vykonávané ako jQuery akcie priradené tlačidlám v elemente *rdt-dialog*. Funkcie a priradené tlačidlá:

- Úprava uzlu
 - *node-edit-button*
- Zmazanie uzlu
 - *node-delete-button*
- Zobrazenie vzťahov uzlu
 - *node-show-relations-button*
- Pridanie nového vzťahu
 - *node-add-relation-button*

Odoslanie úprav

Popis:

Tlačidlu *applyChanges* je priradená funkcia, ktorá prostredníctvom ajax požiadavky.

Vykonanie:

1. Pred odoslaním požiadavky sa zistí, ktoré uzly majú nové pozície. To sa vykoná prechádzaním všetkých prvkov *fd.graph.getNode* a ich porovnávaním s prvotne načítanou premennou *rdts*. Ak je zistená nová pozícia, uzol dostane príznak *newPosition=true*
2. Je nutné potvrdenie používateľa o vykonaní zmien
3. Dáta sa spracujú na servery
4. Je prijatá odpoveď o úspešnom aplikovaní zmien
5. Stránka je znovunačítaná

Spracovanie údajov

Popis:

Spracovanie údajov je vykonávané v modeli variantu.

Použité funkcie:

- `apply_changes`

Vykonanie:

1. Prechádzajú sa dáta získané z *rdtList*
2. Pokiaľ majú v *data.changed* nastavený jeden z príznakov, vykoná sa príslušná akcia:
 - a. `added` – vytvorí sa v databáze nové RDT
 - b. `changed` – zvolené RDT sa upraví
 - c. `deleted` – RDT je odstránené z databázy
 - d. `newPosition` – sú upravené súradnice daného RDT
3. Obdobné operácie sú vykonané aj pre jednotlivých susedov

Prílohy

Príloha A: Obrazovky

V tejto prílohe sú opísané návrhy obrazoviek systému, ktoré dodal zákazník (vedúci projektu) a boli upravované po konzultáciách na tímových stretnutiach.

A.1 Home

Táto obrazovka (obr. 11) sa zobrazí po prihlásení do systému. Obsahuje zoznam všetkých repozitárov, zoznam všetkých variantov metadát a k nim prislúchajúce akcie.

A.2 Repo Detail (RD2)

Na tejto obrazovke (obr. 12) je znázornený detail repozitára. Obsahuje zoznam dokumentov nachádzajúcich sa v repozitári a k nim prislúchajúce akcie. Obsahuje aj zoznam variantov priradených k dokumentom tohto repozitára.

A.3 Metadata Detail – table (V1)

Táto obrazovka (obr. 13) obsahuje detail variantu metadát. Nachádzajú sa tu dve tabuľky. Prvá obsahuje zoznam RDT nachádzajúcich sa v tomto variante, druhá obsahuje vzťahy medzi týmito RDT. Každá tabuľka obsahuje príslušné akcie. Obrazovka obsahuje aj tlačidlo na zobrazenie variantu vo forme grafu (obrazovka VG1).

A.4 Metadata Detail – graph (V1)

Na tejto obrazovke (obr. 14) sa nachádza variant zobrazený vo forme grafu. Vľavo je zoznam RDT, ktoré patria do variantu, vpravo je graf, ktorý zobrazuje vzťahy medzi RDT.

A.5 Repo Detail (RDV1)

V tejto obrazovke (obr. 15) je možné vidieť dokumenty v repozitári a im priradené RDT a vytvárať nové väzby medzi RDT a dokumentami.

A.6 Document Detail – edit (DE1)

Obrazovka DE1 (obr. 16) zobrazuje detail dokumentu v režime editovania. Je na nej editovacie okno pre úpravu obsahu dokumentu. Pod oknom sa nachádzajú akcie pre načítanie obsahu zo súboru, uloženie dokumentu a náhľad dokumentu. Nad editovacím oknom sa nachádza možnosť prepnutia do režimu prezerania dokumentu a možnosť vidieť históriu dokumentu (predchádzajúce verzie). Na pravej strane sa nachádzajú taby s informáciami o dokumente. Tab RDT zobrazuje RDT priradené tomuto dokumentu z rôznych variantov. Tab anotácie zobrazuje anotácie označené v dokumente. Tab súbory obsahuje zoznam súborov (napr. obrázky), ktoré dokument obsahuje. Posledný tab zobrazuje vzťahy dokumentu s ostatnými dokumentmi v repozitári.

A.7 Document Detail – view (DV1)

Obrazovka DV1 (obr. 17) zobrazuje dokument v režime prezerania. V tomto režime sa vedľa dokumentu nachádza tzv. anotačný pásik, na ktorom sú zobrazené označené anotácie v dokumente. Po kliknutí na anotáciu v pásiku je možné anotáciu editovať.

Home

Home

Import...

Export...

Repositories

Name	Version	HasChanged	Documents#
<input type="checkbox"/> Functional Programming	1	yes	297
<input type="checkbox"/> Functional Programming 2012	2	yes	301
<input type="checkbox"/> Logic Programming	5	no	307
<input type="checkbox"/> Software Engineering	1	no	487
<input type="checkbox"/> Software Engineering New	2	yes	490

Create New

Actions
<input type="checkbox"/> <input type="checkbox"/> Publish <input type="checkbox"/> X
<input type="checkbox"/> <input type="checkbox"/> Publish <input type="checkbox"/> X
<input type="checkbox"/> <input type="checkbox"/> Publish <input type="checkbox"/> X
<input type="checkbox"/> <input type="checkbox"/> Publish <input type="checkbox"/> X
<input type="checkbox"/> <input type="checkbox"/> Publish <input type="checkbox"/> X

Popup

Metadata Variants

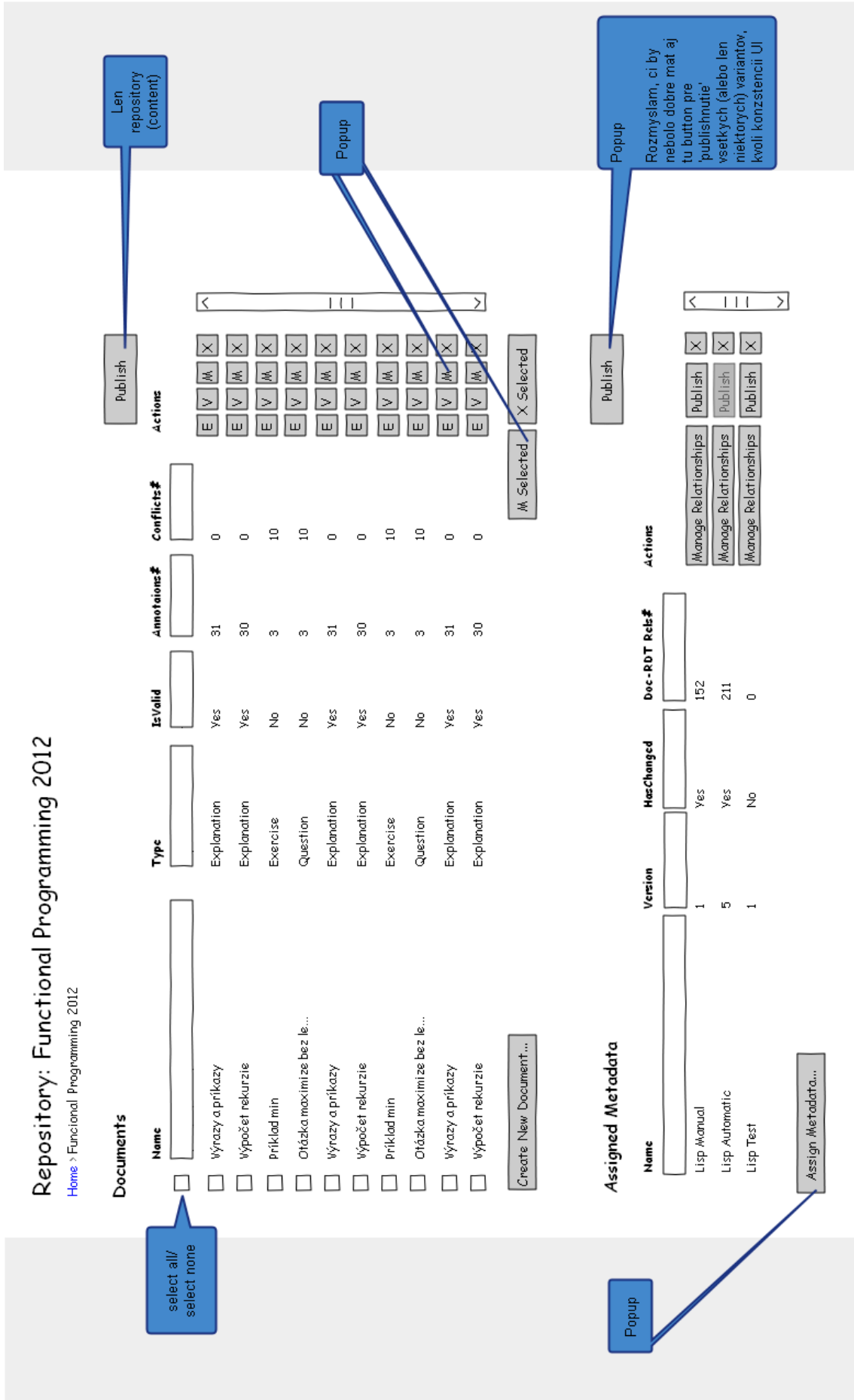
Name	Version	HasChanged	RDT#	RDT Rd#	Repositories#
<input type="checkbox"/> LISp manual	2	yes	301	504	3
<input type="checkbox"/> LISp automatic	1	no	289	578	1
<input type="checkbox"/> LISp new	1	yes	12	0	0
<input type="checkbox"/> PST manual	13	no	531	1003	1
<input type="checkbox"/> PROLOG manual	3	yes	281	423	2
<input type="checkbox"/> PROLOG automatic	2	no	281	399	1
<input type="checkbox"/> PrPr automatic	1	no	471	622	1
<input type="checkbox"/> PrPr manual	1	no	490	640	4

Create New

Actions
<input type="checkbox"/> <input type="checkbox"/> Publish <input type="checkbox"/> X
<input type="checkbox"/> <input type="checkbox"/> Publish <input type="checkbox"/> X
<input type="checkbox"/> <input type="checkbox"/> Publish <input type="checkbox"/> X
<input type="checkbox"/> <input type="checkbox"/> Publish <input type="checkbox"/> X
<input type="checkbox"/> <input type="checkbox"/> Publish <input type="checkbox"/> X
<input type="checkbox"/> <input type="checkbox"/> Publish <input type="checkbox"/> X
<input type="checkbox"/> <input type="checkbox"/> Publish <input type="checkbox"/> X

Popup

Obr. 16. Obrázovka Home.



Obr. 17. Obrazovka Repo Detail (RD2).

Repository/Metadata: FLP-Lisp + Lisp Manual

Home > FLP-Lisp > FLP-Lisp + Lisp Manual

Documents

Name	Type	RDTs#	Actions
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="V"/>
Výrazy a príkazy	<input checked="" type="radio"/> Explanation	5	<input type="button" value="V"/>
Výpočet rekurzie	<input checked="" type="radio"/> Explanation	5	<input type="button" value="V"/>
Príklad min	<input checked="" type="radio"/> Exercise	15	<input type="button" value="V"/>
Otázka maximálne bezle...	<input checked="" type="radio"/> Question	10	<input type="button" value="V"/>
Výrazy a príkazy	<input checked="" type="radio"/> Explanation	0	<input type="button" value="V"/>
Výpočet rekurzie	<input checked="" type="radio"/> Explanation	0	<input type="button" value="V"/>
Príklad min	<input checked="" type="radio"/> Exercise	10	<input type="button" value="V"/>
Otázka maximálne bezle...	<input checked="" type="radio"/> Question	3	<input type="button" value="V"/>
Výrazy a príkazy	<input checked="" type="radio"/> Explanation	31	<input type="button" value="V"/>
Výpočet rekurzie	<input checked="" type="radio"/> Explanation	0	<input type="button" value="V"/>
Výrazy a príkazy	<input checked="" type="radio"/> Explanation	5	<input type="button" value="V"/>
Výpočet rekurzie	<input checked="" type="radio"/> Explanation	5	<input type="button" value="V"/>
Príklad min	<input checked="" type="radio"/> Exercise	15	<input type="button" value="V"/>
Otázka maximálne bezle...	<input checked="" type="radio"/> Question	10	<input type="button" value="V"/>
Výrazy a príkazy	<input checked="" type="radio"/> Explanation	0	<input type="button" value="V"/>
Výpočet rekurzie	<input checked="" type="radio"/> Explanation	0	<input type="button" value="V"/>
Príklad min	<input checked="" type="radio"/> Exercise	10	<input type="button" value="V"/>
Otázka maximálne bezle...	<input checked="" type="radio"/> Question	3	<input type="button" value="V"/>
Výrazy a príkazy	<input checked="" type="radio"/> Explanation	31	<input type="button" value="V"/>
Výpočet rekurzie	<input checked="" type="radio"/> Explanation	0	<input type="button" value="V"/>
Výrazy a príkazy	<input checked="" type="radio"/> Explanation	5	<input type="button" value="V"/>
Výpočet rekurzie	<input checked="" type="radio"/> Explanation	5	<input type="button" value="V"/>
Príklad min	<input checked="" type="radio"/> Exercise	15	<input type="button" value="V"/>
Otázka maximálne bezle...	<input checked="" type="radio"/> Question	10	<input type="button" value="V"/>
Výrazy a príkazy	<input checked="" type="radio"/> Explanation	0	<input type="button" value="V"/>
Výpočet rekurzie	<input checked="" type="radio"/> Explanation	0	<input type="button" value="V"/>
Príklad min	<input checked="" type="radio"/> Exercise	10	<input type="button" value="V"/>
Otázka maximálne bezle...	<input checked="" type="radio"/> Question	3	<input type="button" value="V"/>
Výrazy a príkazy	<input checked="" type="radio"/> Explanation	31	<input type="button" value="V"/>
Výpočet rekurzie	<input checked="" type="radio"/> Explanation	0	<input type="button" value="V"/>

Preview popup, to oko možno môže byť aj na začiatku riadka alebo hneď za názvom (Name)

Ten prvý stĺpec zatiaľ nedávať

Assigned Metadata: Výrazy a príkazy

From	To	Type	Weight	Actions
<input type="checkbox"/> Výrazy a príkazy	<input type="checkbox"/> MAPCAR	<input type="checkbox"/> has-ndt	0.33	<input type="button" value="E"/> <input type="button" value="X"/>
<input type="checkbox"/> Funkcionály v LISpe	<input type="checkbox"/> funkcionál	<input type="checkbox"/> has-ndt	1.00	<input type="button" value="E"/> <input type="button" value="X"/>
<input type="checkbox"/> Čistý výraz	<input type="checkbox"/> výraz	<input type="checkbox"/> has-ndt	0.90	<input type="button" value="E"/> <input type="button" value="X"/>
<input type="checkbox"/> Čistý výraz	<input type="checkbox"/> príkaz	<input type="checkbox"/> has-ndt	0.70	<input type="button" value="E"/> <input type="button" value="X"/>
<input checked="" type="checkbox"/> Čistý výraz	<input type="checkbox"/> rekurzia	<input type="checkbox"/> has-ndt	0.05	<input type="button" value="E"/> <input type="button" value="X"/>
<input type="checkbox"/> Výrazy a príkazy	<input type="checkbox"/> MAPCAR	<input type="checkbox"/> has-ndt	0.33	<input type="button" value="E"/> <input type="button" value="X"/>
<input type="checkbox"/> Funkcionál MAPCAR	<input type="checkbox"/> funkcionál	<input type="checkbox"/> has-ndt	1.00	<input type="button" value="E"/> <input type="button" value="X"/>
<input type="checkbox"/> Funkcionál MAPCAR	<input type="checkbox"/> MAPCAR	<input type="checkbox"/> has-ndt	0.95	<input type="button" value="E"/> <input type="button" value="X"/>
<input type="checkbox"/> Vetvenie	<input type="checkbox"/> COND	<input type="checkbox"/> has-ndt	0.81	<input type="button" value="E"/> <input type="button" value="X"/>
<input type="checkbox"/> Vetvenie	<input type="checkbox"/> forma	<input type="checkbox"/> has-ndt	0.57	<input type="button" value="E"/> <input type="button" value="X"/>
<input checked="" type="checkbox"/> Výrazy a príkazy	<input type="checkbox"/> MAPCAR	<input type="checkbox"/> has-ndt	0.33	<input type="button" value="E"/> <input type="button" value="X"/>
<input type="checkbox"/> Funkcionály v LISpe	<input type="checkbox"/> funkcionál	<input type="checkbox"/> has-ndt	1.00	<input type="button" value="E"/> <input type="button" value="X"/>
<input type="checkbox"/> Čistý výraz	<input type="checkbox"/> výraz	<input type="checkbox"/> has-ndt	0.90	<input type="button" value="E"/> <input type="button" value="X"/>
<input type="checkbox"/> Čistý výraz	<input type="checkbox"/> príkaz	<input type="checkbox"/> has-ndt	0.70	<input type="button" value="E"/> <input type="button" value="X"/>
<input type="checkbox"/> Čistý výraz	<input type="checkbox"/> rekurzia	<input type="checkbox"/> has-ndt	0.05	<input type="button" value="E"/> <input type="button" value="X"/>
<input type="checkbox"/> Výrazy a príkazy	<input type="checkbox"/> MAPCAR	<input type="checkbox"/> has-ndt	0.33	<input type="button" value="E"/> <input type="button" value="X"/>
<input type="checkbox"/> Funkcionál MAPCAR	<input type="checkbox"/> funkcionál	<input type="checkbox"/> has-ndt	1.00	<input type="button" value="E"/> <input type="button" value="X"/>
<input type="checkbox"/> Funkcionál MAPCAR	<input type="checkbox"/> MAPCAR	<input type="checkbox"/> has-ndt	0.95	<input type="button" value="E"/> <input type="button" value="X"/>
<input type="checkbox"/> Vetvenie	<input type="checkbox"/> COND	<input type="checkbox"/> has-ndt	0.81	<input type="button" value="E"/> <input type="button" value="X"/>
<input type="checkbox"/> Vetvenie	<input type="checkbox"/> forma	<input type="checkbox"/> has-ndt	0.57	<input type="button" value="E"/> <input type="button" value="X"/>

select all/
select none

inline

autocomplete
combobox

X Selected

Obr. 20. Obrázovka Repo Detail (RDV1).

Document-ID: flip-book-p-2.2

Abstrakčný typ údajov lisp-zoznam

Praktické používanie typov údajov si vyžaduje ich presný opis - špecifikáciu. Nestačí iba vymenovať príпустné operácie, povolené spôsoby ich kombinácií do formulí (signatúra). Treba opísať aj ich sémantiku. Na opis sémantiky sa často používa *axiomatizovaná špecifikácia*. V axiomatizkej špecifikácii sa sémantika operácií opisuje logickým formulami - axiómami. Z hľadiska typu údajov axiómy vyjadrujú podstatné vlastnosti operácií, a to nezávisle od toho, ako sa bude typ údajov implementovať. V axiomatizkej špecifikácii sa teda sústreďujeme na to, čo jednotlivé operácie robia a robí sa abstrakcia od toho, AKO to robia, ako sa štruktúry údajov reprezentujú a ako sa realizujú operácie. Hovoríme preto o *abstrakčných typoch údajov* (angl. abstract data type - ADT).

Abstrakčný typ údajov zahŕňa hodnoty a elementárne operácie definované nad týmito hodnotami. Najsťor budeme uvádzať hodnoty typu lisp-zoznam, a potom aj operácie nad týmito hodnotami. Vzáťahom na to, že zoznam v lisp-e je špeciálny prípad abstrakčného typu údajov zoznamu, označili sme ho pojmom *lisp-zoznam*. V ďalšom texte budeme pod pojmom *zoznam myslieť vždy lisp-zoznam*. V prípade potreby túto skutočnosť explicitne zdôrazníme. Zároveň budeme na ...

The screenshot shows a document editor interface. At the top, there are buttons for 'Edit' and 'View'. Below that, a toolbar contains 'Properties?', 'Annotations', 'RD-Ts', 'Document Relationships', and 'Files'. The main content area displays a table with columns for 'SelectedText/Context', 'Content', 'Type', 'Author', and 'HasCont Actions'. The table contains several rows of data, including entries for 'Model umožňuje', 'sysgst', 'Softvérové procesy', and 'Model umožňuje'. A context menu is open over the first row, showing options like 'Míchal Búčka', 'toto označenie sa mne osobne ve...', 'Alžbeta Tichá', 'v texte je chyba', and 'István Szonodyi', each with 'E' and 'X' icons.

SelectedText/Context	Content	Type	Author	HasCont Actions
Model umožňuje	Model umožňuje	Highlight	Milan Stručík	No
sysgst	V tomto texte je preklep	Error report	Agáda Impl...	Yes
Identifikátory, napr.: systst a systst	V židnem prípade	Tag	Peter Čečka	No
2 Softvérové procesy	V tejto časti sa venujeme opisu softvérových	Tag	Peter Čečka	No
Model umožňuje	Model umožňuje	Highlight	Milan Stručík	No
sysgst	V tomto texte je preklep	Error report	Agáda Impl...	Yes
Identifikátory, napr.: systst a systst	V židnem prípade	Tag	Peter Čečka	No
2 Softvérové procesy	V tejto časti sa venujeme opisu softvérových	Tag	Peter Čečka	No

Neviem, čo s properties, ako Name, LO-ID, atd. Zbytočne zabierajú miesto nad obsahom súboru, možno by som ich dal ako samostatnú tabku 'Properties'

Popup

Obr. 22. Obrázovka Document Detail – view (DV1).

A.8 Tabuľky v systéme

Systém obsahuje niekoľko zobrazení vo forme tabuliek. Tu je uvedený ich zoznam s požadovanými funkcionalitami a obsahom.

Obrazovka	Tabuľka-Názov	Stĺpce	Akcie-Riadok	Akcie-Tabuľka	Typ
Home	Repositories	Name, #Documents, Actions	Show (RD2), Edit Properties, Delete	Add, Import, Export	Sortable, Scrollable
Home	Metadata	Name, #RDTs, #RDT Relationships, #Repositories, Actions	Show (V1), Edit Properties, Delete	Add, Import, Export	Sortable, Scrollable
RD2	Documents	Name, Valid, #RDTs, #Annotations, #Conflicts, Actions	Show (DP1), Show-Edit (DE1), Edit Properties, Move, Delete	Move, Delete	Sortable, Selectable, Filterable, Scrollable
RD2	Assigned Metadata	Name, #Document-RDT Relationships, Actions	Show, Edit Relationships, Remove	Add	-
DE1	Document RDTs	Name, Type, Weight, Actions	Edit, Remove	Add	Pivot – Metadata, Sortable, Selectable, Filterable, Scrollable
DE1	Document Annotations	Text + Selection	-	-	Sortable, Filterable
DE1	Document Files	Filename, Size, Actions	Remove	Choose File, Add	-
DE1	Document Relationships	Name, Type, Weight, Actions	Remove	Add	Sortable, Filterable, Scrollable
V1	RDTs	Name, Actions	Edit, Delete	Add, Delete	Sortable, Selectable, Filterable, Scrollable
V1	Relationships	From, To, Type, Weight, Actions	Edit, Delete	Add, Delete	Sortable, Selectable, Filterable, Pageable

Pozn.: V zátvorke je odkaz na obrazovku, kam sa dostať. Nie je to súčasť názvu akcie.