

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
e-mail: tim13@googlegroups.com
Ak. rok : 2011/2011

Obsah

1	Úvod	1-1
1.1	Použité pojmy	1-2
2	Šprint č. 5 – Elgar	2-1
2.1	Ohraničenie zobrazovania vrcholov	2-1
2.2	Kontrola integrity dokumentu.....	2-2
2.3	Obrázky pri transformácii.....	2-4
2.4	Efektívne priradovanie RDT k dokumentom	2-4
3	Šprint č. 6 – Fischer	3-1
3.1	Komunikácia Fely s klientmi	3-1
3.2	Rozšírenie modelu používateľa	3-4
3.3	Výber vzťahov medzi RDT podľa vybraného RDT (obrazovka V1)	3-6
3.4	Tabuľka susedov vrcholu nad grafom	3-7
3.5	Vizualizácia uzlov ako textových	3-8
3.6	Filtrovanie tabuliek na úrovni stĺpcov	3-9
4	Šprint č. 7 – Grieg.....	4-1
4.1	Preview dokumentu pri akcii hover v obrazovke RDV1	4-1
4.2	"Advanced" tab na zobrazenie anotácií	4-3
4.3	Pop-up v grafe.....	4-3
4.4	Graf property panel.....	4-4
4.5	Komunikácia systému s klientmi – Obsah.....	4-6
5	Šprint č. 8 – Haydn.....	5-1
5.1	Komunikácia systému s klientmi – Anotácie.....	5-1
5.2	Edit anotácie	5-3
5.3	Vkladanie anotácií podľa ALEFu	5-4
5.4	Import repozitára + metadát: infobox s výberom	5-5
5.5	Zapamätanie si pozícií vrcholov v grafe.....	5-7
6	Opis prototypu	6-1
6.1	Funkcionalita systému	6-1
6.2	Architektúra systému	6-8
6.3	Dátový model systému.....	6-9

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.

Dokumentácia obsahuje viacero častí, na ktorých sa autorsky podieľali jednotliví členovia tímu nasledovne:

Úvod Bc. Martin Gajdoš

Šprint č. 5 – Elgar

Ohraničenie zobrazovania vrcholov	Bc. Martin Gajdoš
Kontrola integrity dokumentu	Bc. Martin Franta
Obrázky pri transformácii	Bc. Martin Habdák
Efektívne priradovanie RDT k dokumentom	Bc. Martin Franta

Šprint č. 6 – Fischer

Komunikácia Fely s klientmi	Bc. Martin Franta
Rozšírenie modelu používateľa	Bc. Martin Habdák
Výber vzťahov medzi RDT podľa vybraného RDT (V1)	Bc. Petra Vrabecová
Tabuľka susedov vrcholu nad grafom	Bc. Petra Vrabecová
Vizualizácia uzlov ako textových	Bc. Martin Gajdoš
Filtrovanie tabuliek na úrovni stĺpcov	Bc. Róbert Kocian

Šprint č. 7 – Grieg

Preview dokumentu pri akcii hover v RDV1	Bc. Martin Habdák
"Advanced" tab na zobrazenie anotácií	Bc. Róbert Kocian
Pop-up v grafe	Bc. Martin Gajdoš
Graf property panel	Bc. Petra Vrabecová
Komunikácia systému s klientmi – Obsah	Bc. Martin Franta

Šprint č. 8 – Haydn

Komunikácia systému s klientmi – Anotácie	Bc. Martin Franta
Edit anotácie	Bc. Petra Vrabecová

Vkladanie anotácií podľa ALEFu	Bc. Martin Gajdoš
Import repozitára + metadát: infobox s výberom	Bc. Martin Habdák
Zapamätanie si pozícií vrcholov v grafe	Bc. Martin Gajdoš

Opis prototypu

Funkcionalita systému	Bc. Martin Gajdoš, Bc. Róbert Kocian, Bc. Petra Vrablecová
Architektúra systému	Bc. Martin Franta
Dátový model	Bc. Martin Habdák

Prílohy

Obrazovky	Bc. Petra Vrablecová
-----------	----------------------

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 – priečinkov 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 Šprint č. 5 – Elgar

2.1 Ohraničenie zobrazovania vrcholov

Úlohou bolo pridanie funkcionality, ktorá by používateľovi poskytla možnosť, ako ohraničiť zobrazovanie susedov od daného uzlu, do ním požadovanej hĺbky.

2.1.1 Použitie

1. Používateľ v grafovom okne zobrazí možnosti stlačením položky *Options*
2. Nastaví hĺbku do ktorej požaduje zobrazenie obmedziť v sekcii *Neighbourhood size*
3. Zvolí uzol od ktorého má byť ohraničenie vykonané

2.1.2 Implementácia

Vytvorená funkcia `constrainNode` v súbore `assets/javascripts/graph_view.js`.

Ako parametre berie uzol, od ktorého má byť ohraničenie vykonané. V jej tele je použitá globálna premenná `constraintEnabled`, ktorá udáva, či je ohraničovanie zapnuté. Hĺbku do ktorej má byť ohraničovanie vykonané zisťuje z elementu `depth` typu `slider`.

Funkcia je spúšťaná vždy pri výbere uzlu v rámci grafu.

2.1.3 Testovanie

Názov testu		Test ohraničenia vrcholov	ID	2-1-1
Rozhranie	Okno grafu V1		autor	Martin Gajdoš
Účel	Otestovanie funkcionality			
Vstupné podmienky		Načítaný graf.		
Výstupné podmienky		Ohraničené zobrazenie grafu.		
Krok	Akcia	Očakávaná akcia	Skutočná akcia	
1	Zvolíme možnosť <i>Options</i> .	Zobrazené <i>Options</i> .	Zobrazené <i>Options</i> .	
2	Vybranie veľkosti ohraničenia 3.	Nastavenie ohraničenia na 3.	Nastavenie ohraničenia na 3.	
3	Výber uzlu.	Ohraničenie zobrazenia do hĺbky 3 od uzlu.	Ohraničenie zobrazenia do hĺbky 3 od uzlu.	

2.2 Kontrola integrity dokumentu

Kontrola, či:

- ▲ sú priložené obrázky referencované v zdrojovom XML,
- ▲ anotácie nepresahujú obsah dokumentu.

1.1.1 Analýza

Validácia dokumentu sa vzťahuje iba na správnu štruktúru XML. Používateľ potrebuje pri ukladaní dokumentu zistiť, či je zachovaná jeho integrita. Úpravami je možné zmeniť dokument tak, že anotácie presahujú obsah, čo doposiaľ nebolo predmetom overovania.

Súbory priložené k dokumentu sú linkované pomocou XML elementov <imagedata>. Bude preto potrebné kontrolovať, či súhlasia imagedata elementy v dokumente so súbormi, ktoré používateľ k súboru priložil (prípadne boli vložené pri importe repozitára).

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

1.1.2 Návrh

Kontrola integrity bude pri uložení dokumentu realizovaná dvomi testami:

- ▲ porovnaním začiatkových a koncových súradníc anotácie s počtom znakov dokumentu,
- ▲ vyhľadáním názvov súborov z jednotlivých imagedata elementov medzi súbormi, ktoré sú k dokumentu priložené.

Následne bude používateľ v chybovom výpise upozornený na chýbajúce súbory a nesprávne umiestnené anotácie. Testy sa vykonávajú, ak vo formulári pre úpravu, resp. vytvorenie dokumentu bude označené zaškrŕavacie pole určujúce, či používateľ požaduje kontrolu integrity.

2.2.1 Implementácia

Do akcie Update v súbore DocumentsController bola pridaná podmienka zisťujúca, či bolo označené zaškrŕavacie pole vo formulári dokumentu. Na jej základe sa volá metóda check_integrity z modelu Document. Táto metóda získava informáciu o presahujúcich anotáciách z metódy check_annotations, ktorá slúži na zisťovanie konfliktov v anotáciách. V metóde check_annotations sa postupne prechádzajú anotácie prislúchajúce k dokumentu a zisťuje sa ich rozsah vzhľadom k dĺžke dokumentu. Po zistení chybnjej anotácie sa nastaví príznak range_conflict na hodnotu true a do poľa errors je pridaná informácia o chybnjej anotácii. Pole errors obsahuje chybové informácie z validácie dokumentu, ktoré sa používateľovi zobrazia pri neúspešnom uložení dokumentu. Takto získa prehľad, ktoré anotácie už nezodpovedajú aktuálnemu stavu dokumentu.

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. Uvedený postup sa vykonáva v metóde check_integrity v modeli Document.

2.2.2 Testovanie

Názov testu		Nájdenie presahujúcej anotácie	ID	2-2-1
Rozhranie	Zobrazenie dokumentu		autor	Martin Franta
Účel	Zaistenie integrity dokumentu a anotácií			
Vstupné podmienky	Dokument je vytvorený a je validný.			
Výstupné podmienky	Chybový výpis obsahuje vytvorenú anotáciu.			
Krok	Akcia	Očakávaná akcia	Skutočná akcia	
1	Pridáme anotáciu, ktorá končí na poslednom znaku dokumentu.	Anotácia je vytvorená.	Anotácia bola vytvorená.	
2	Upravíme dokument tak, že odstránime posledné znaky textu dokumentu, označíme kontrolu integrity a uložíme.	Zobrazí a chybový výpis s vytvorenou anotáciou.	Zobrazil sa chybový výpis s vytvorenou anotáciou.	

Názov testu		Upozornenie na chýbajúci súbor	ID	2-2-2
Rozhranie	Formulár editácie dokumentu		autor	Martin Franta
Účel	Zaistenie integrity dokumentu a príloh			
Vstupné podmienky	Dokument je vytvorený a je validný.			
Výstupné podmienky	Chybový výpis obsahuje názov chýbajúceho súboru.			
Krok	Akcia	Očakávaná akcia	Skutočná akcia	
1	Do obsahu dokumentu vložíme tag <imagedata format="png" fileref="testpicture.png"/>, povolíme kontrolu integrity a uložíme.	Zobrazí sa chybový výpis s názvom súboru testpicture.png.	Zobrazil sa chybový výpis s názvom súboru testpicture.png.	

2.3 Obrázky pri transformácii

Pri zobrazení transformácie chcem vidieť zobrazené aj obrázky priradené k dokumentu.

2.3.1 Analýza

Dokumenty sú v systéme uložené vo formáte XML. Obrázok v dokumente je referencovaný pomocou atribútu, ktorý ukazuje na požadovaný súbor. Ukladanie priložených obrázkov sa vykonáva gemom „Paperclip“. Spomenutý gem ukladá obrázky s iným názvom, aký je uvedený v atribúte elementu obrázka v tele dokumentu. Iný spôsob ukladania je implementovaný z toho dôvodu, aby boli obrázky jednoznačne identifikované podľa ID a mohli mať rovnaké názvy. Z dôvodu iného mena obrázka nedochádza k automatickému prepojeniu na obrázok a následne nie je zobrazený po transformácii do formátu HTML.

2.3.2 Návrh

Pri importovaní je možné zmeniť obsah dokumentu tak, aby atribúty obrázkových elementov referencovali správne súbory. Zmenia sa teda na takú hodnotu, pod ktorou je obrázok skutočne uložený v systéme. Pri transformácii sa správne naviaže referencia na obrázok a ten sa následne vloží do vytvoreného HTML dokumentu.

2.3.3 Implementácia

Pri importovaní dokumentov vo formáte XML sa prezrie jeho obsah a získa sa zoznam všetkých obrázkov, ktoré sa v ňom nachádzajú. Obrázky sa uložia v systéme a získa sa ich identifikátor. Následne sa zmenia atribúty elementov obrázkov v tele dokumentu. Po transformácii sa vytvoria správne referencie na skutočný obrázok. Názov obrázku a teda aj atribútu príslušného elementu je odvodený od ID obrázka v systéme, ku ktorému je pripojený reťazec „_photo“.

2.4 Efektívne priradovanie RDT k dokumentom

Vytvorenie novej obrazovky, ktorá bude obsahovať dve prepojené tabuľky: dokumenty a priradené RDT.

2.4.1 Analýza

Používateľ chce vybrať v tabuľke dokument, pre ktorý sa asynchrónne do tabuľky priradených RDT načítajú všetky RDT prislúchajúce k danému dokumentu. Zoznam RDT bude možné upravovať, pridávať nové vzťahy RDT – dokument, vymazávať vzťahy a upravovať atribúty vzťahu.

Práca so vzťahmi musí byť efektívna – minimalizuje sa počet akcií, ktoré používateľ potrebuje vykonať aby dosiahol požadovaný výsledok. Tomu zodpovedá asynchrónne načítanie riadkov do tabuľky po označení dokumentu a editácia vzťahu priamo v riadku tabuľky.

1.1.3 Návrh

Obrazovka bude prístupná zo zoznamu variantov priradených k repozitáru. K dokumentom v danom repozitári bude možné priradovať RDT z vybraného variantu.

Označením dokumentu v tabuľke sa zobrazia RDT, ktoré sú už priradené. Tabuľka dokumentov bude tiež obsahovať odkaz na zobrazenie dokumentu, počet priradených RDT a možnosť rýchleho náhľadu dokumentu. V tabuľke bude možné označiť iba jeden riadok.

Tabuľka s priradenými RDT bude obsahovať názov RDT, typ vzťahu a jeho váhu, akciu vymazania a úpravy vzťahu. Používateľ môže v tabuľke označiť viacero riadkov, čo umožní spustenie akcie hromadného vymazania vzťahov.

Pridanie vzťahu bude zabezpečené formulárom, v ktorom používateľ vyberie RDT, typ vzťahu a jeho váhu. Výber RDT pre nový vzťah bude realizovaný rolovacím menu s funkciou autocomplete.

1.1.4 Implementácia

Tabuľky dokumentov a ich vzťahov s RDT boli upravené pre použitie s JavaScript knižnicou Datables, ktorá umožňuje filtráciu a zoradovanie dát v tabuľke. Pri zobrazovaní vzťahov bolo potrebné zabezpečiť dynamické pridávanie dát do tabuľky po asynchrónnej požiadavke vyvolanej označením dokumentu. Túto vlastnosť tiež zabezpečila funkcia knižnice Datables, čím sa pridané riadky okamžite stali súčasťou gridu.

Úprava vzťahu v riadku tabuľky bola implementovaná vložením formulára po tom, ako používateľ klikne na prvok vyvolávajúci akciu Edit. Potvrdením zmien vo formulári sa odošle asynchrónna požiadavka a dáta v tabuľke sa po jej úspešnom vybavení aktualizujú.

Zrušenie vzťahu je tiež realizované asynchrónnou požiadavkou. Následne sa aktualizuje tabuľka vzťahov a počet RDT priradených k dokumentu.

Hromadná akcia zrušenia vybraných vzťahov je spojením viacerých akcií pre jednotlivé vymazanie vzťahu.

Pridanie vzťahu k vybranému dokumentu je asynchrónna požiadavka vyvolaná odoslaním potvrdením formuláru pre nový vzťah. Rolovacie menu pre výber RDT je naplnené všetkými RDT, ktoré daný variant obsahuje. Funkcia autocomplete je zabezpečená knižnicou jQuery UI.

Uvedená implementácia sa nachádza v súboroch rdt_document_relationships.js.coffee, v controlleri RdtDocumentRelationshipsController a v príslušných šablónach.

2.4.2 Testovanie

Názov testu		Zobrazenie vzťahov	ID	2-4-1
Rozhranie	Obrazovka RDV1 (vzťahy RDT - dokument pre daný repozitár a variant metadát)		autor	Martin Franta
Účel	Získanie vzťahov medzi dokumentom a RDT z daného variantu			
Vstupné podmienky		Repozitár obsahuje dokumenty s priradenými RDT.		
Výstupné podmienky		Zobrazené sú priradené RDT k vybranému dokumentu.		
Krok	Akcia	Očakávaná akcia	Skutočná akcia	
1	Kliknutím vyberieme dokument v tabuľke s počtom priradených RDT > 0.	V hornej časti obrazovky sa zobrazí názov dokumentu, v tabuľke v pravej časti sa načítajú RDT priradené k dokumentu.	V hornej časti obrazovky sa zobrazil názov dokumentu, v pravej časti sa načítali RDT priradené k dokumentu.	

Názov testu		Úprava vzťahu	ID	2-4-2
Rozhranie	Obrazovka RDV1 (vzťahy RDT - dokument pre daný repozitár a variant metadát)		autor	Martin Franta
Účel	Úprava vzťahu medzi dokumentom a RDT z daného variantu			
Vstupné podmienky	Repozitár obsahuje dokument s priradenými RDT.			
Výstupné podmienky	Vzťah medzi dokumentom a RDT je upravený.			
Krok	Akcia	Očakávaná akcia	Skutočná akcia	
1	Kliknutím na riadok tabuľky vyberieme dokument v tabuľke s počtom priradených RDT > 0.	V hornej časti obrazovky sa zobrazí názov dokumentu, v tabuľke v pravej časti sa načítajú RDT priradené k dokumentu.	V hornej časti obrazovky sa zobrazil názov dokumentu, v pravej časti sa načítali RDT priradené k dokumentu.	
2	Klikneme na akciu Edit v riadku zvoleného vzťahu.	Zobrazia sa formulárové polia pre úpravu váhy a typu vzťahu.	Zobrazili sa formulárové polia pre úpravu váhy a typu vzťahu.	
3	Upravíme váhu na hodnotu z intervalu (0, 1> odlišnú od aktuálnej hodnoty a vzťah kliknutím na akciu Save (Uložiť) uložíme.	Formulárové polia sú nahradené textom, váha má novú hodnotu.	Formulárové polia boli nahradené textom, váha mala novú hodnotu.	

Názov testu		Vytvorenie vzťahu	ID	2-4-3
Rozhranie	Obrazovka RDV1 (vzťahy RDT - dokument pre daný repozitár a variant metadát)		autor	Martin Franta
Účel	Vytvorenie vzťahu medzi dokumentom a RDT z daného variantu			
Vstupné podmienky	Repozitár obsahuje dokument, variant obsahuje RDT.			
Výstupné podmienky	Vzťah medzi dokumentom a RDT je vytvorený.			
Krok	Akcia	Očakávaná akcia	Skutočná akcia	
1	Kliknutím na riadok tabuľky vyberieme dokument v tabuľke.	V hornej časti obrazovky sa zobrazí názov dokumentu.	V hornej časti obrazovky sa zobrazil názov dokumentu.	

2	V spodnej časti obrazovky vyberieme RDT pre nový vzťah, jeho typ a váhu z intervalu (0, 1>. Potvrdíme vytvorenie vzťahu.	Do tabuľky priradených RDT pribudne nový riadok s hodnotami, ktoré sme zadali do formulára.	V tabuľke pribudol nový riadok s hodnotami, ktoré sme zadali do formulára.
---	--	---	--

Názov testu		Zrušenie vzťahu	ID	2-4-4
Rozhranie	Obrazovka RDV1 (vzťahy RDT - dokument pre daný repozitár a variant metadát)		autor	Martin Franta
Účel	Zrušenie vzťahu medzi dokumentom a RDT z daného variantu			
Vstupné podmienky	Repozitár obsahuje dokument s priradeným RDT.			
Výstupné podmienky	Vzťah medzi dokumentom a RDT je zrušený.			
Krok	Akcia	Očakávaná akcia	Skutočná akcia	
1	Kliknutím na riadok tabuľky vyberieme dokument v tabuľke.	V hornej časti obrazovky sa zobrazí názov dokumentu	V hornej časti obrazovky sa zobrazil názov dokumentu.	
2	V tabuľke priradených RDT klikneme na akciu Delete (Vymazať).	Zobrazí sa dialóg pre potvrdenie akcie.	Zobrazil sa dialóg pre potvrdenie akcie.	
3	Vyberieme možnosť Áno (Ok/Yes).	Vzťah je odstránený z tabuľky, počet RDT v riadku dokumentu sa zníži o 1.	Vzťah bol odstránený z tabuľky, počet RDT v riadku dokumentu sa znížil o 1.	

3 Šprint č. 6 – Fischer

3.1 Komunikácia Fely s klientmi

Cieľom je oboznámiť klientov systému o zmenách v metadátach (RDT, vzťahy RDT-RDT a RDT-dokument).

3.1.1 Analýza

Analýzou požadovanej komunikácie medzi systémom FELA a klientmi (navrhované pre ALEF), boli identifikované nasledujúce scenáre:

I. Opis: Chceme vytvoriť nový setup v ALEFe a priradiť mu obsah a metadáta vo FELA.

Akcia:

1. Pri tvorbe nového setupu si ALEF vyžiada všetky repozitáre, ktoré spravuje FELA.
2. FELA pošle zoznam repozitárov s priradenými metadátami.

II. Opis: Chceme editovať metadáta alebo vzťahy medzi dokumentmi a RDT.

Akcia:

1. Prihlásim sa do FELA a robím zmeny.
2. Po dokončení zmien kliknem na `↑ Publish ↓`, čím sa metadáta stanú dostupné pre ALEF.

III. Opis: Vieme, že FELA obsahuje novú verziu metadát pre setup a chceme to nahráť do ALEFu.

Akcia:

1. Stlačíme button `↑ Update ↓` (metadata) pre získanie verzie metadát.
2. FELA pošle metadáta, ktorými sa nahradia metadáta v danom setepe v ALEFe.

Rozdiel medzi verziou metadát vo FELA a u klientov určuje, či v metadátach došlo k zmene na strane systému FELA.

3.1.2 Návrh

Zmena verzie metadát nastane vyvolaním akcie Publish na strane systému FELA. Akciu publish bude možné vykonať iba nad zmenenými metadátami, čo určuje príznak `has_changed` s hodnotou `true`.

K zmene príznaku `has_changed` pre variant metadát dochádza pri zmene RDT, vzťahu RDT-RDT a samotného variantu.

Zároveň systém pozná priradenia variantov k repozitárom, ktoré sú tiež týmito zmenami ovplyvnené. Preto každá zmena variantu nastaví príznak `has_changed` aj všetkým väzbám tohto variantu na repozitáre. Zmenu väzby tiež vyvoláva zmena vzťahu RDT-dokument, a to zmenu práve tej väzby, ktorá spája variant s daným RDT s repozitárom obsahujúcim daný dokument.

Akciou publish sa verzia zvyšuje a príznak `has_changed` je opäť nastavený na hodnotu `false`.

Pre získanie zoznamu repozitárov s metadátami bol navrhnutý formát JSON odpovede:

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

Odpoveďou na požiadavku z komunikácie III. je zip archív metadát, ktorého štruktúra zodpovedá štruktúre metadát v archíve exportu repozitára.

3.1.3 Implementácia

Funkcionalita bola implementovaná pomocou callback funkcií modelov Rdt, RdtsRelationship a RdtDocumentRelationship. Callback funkcia je vyvolaná po uložení modelu a zmení príznak `has_changed` buď priamo v modeli, v ktorom bola vyvolaná, alebo v modeli nadradenej entity (napr. Rdt zmení Variant).

Publish funkcia je vyvolaná z controllera na daný model (Variant, RepositoryVariant), čím sa zmení atribút modelu `version` a príznak `has_changed` je nastavený na `false`.

Bola vytvorená akcia List, ktorej odpoveďou je zoznam repozitárov s metadátami vo formáte JSON.

Akcia `export` v controlleri `VariantsController` využíva už implementovanú funkciu `Export` pre repozitáre s tým rozdielom, že exportované sú iba samotné metadáta.

3.1.4 Testovanie

Názov testu		Zmena verzie variantu metadát	ID	3-1-1
Rozhranie	Detail variantu – tabuľkové zobrazenie		autor	Martin Franta
Účel	Oboznámenie klientov o zmenách vo variante metadát			
Vstupné podmienky	Variant obsahuje RDT, jeho príznak has_changed je false.			
Výstupné podmienky	Verzia variantu je zmenená, jeho príznak has_changed je false.			
Krok	Akcia	Očakávaná akcia	Skutočná akcia	
1	V tabuľke RDT upravíme názov zvoleného RDT.	RDT je zmenený.	RDT bolo zmenený.	
2	Na úvodnej obrazovke v časti Variants klikneme na aktívny odkaz Publish pre zvolený variant.	Odkaz Publish sa deaktivuje, verzia sa zvýši, has_changed zmení hodnotu na false.	Odkaz Publish sa deaktivoval, verzia sa zvýšila, has_changed zmenil hodnotu na false.	

Názov testu		Zmena verzie väzby variant - repozitár	ID	3-1-2
Rozhranie	Obrazovka RDV1 – editácia vzťahov RDT-dokument		autor	Martin Franta
Účel	Oboznámenie klientov o zmenách vo väzbe variant - repozitár			
Vstupné podmienky	Existuje vzťah medzi dokumentom a RDT.			
Výstupné podmienky	Verzia väzby variantu a repozitára je zmenená, jeho príznak has_changed je false.			
Krok	Akcia	Očakávaná akcia	Skutočná akcia	
1	V tabuľke dokumentov vyberieme dokument.	V tabuľke vzťahov sa zobrazia RDT priradené k dokumentu.	V tabuľke vzťahov sa zobrazili RDT priradené k dokumentu.	
2	Upravíme vzťah RDT-dokument zmenou váhy vzťahu.	Vzťah RDT-dokument bude mať zmenenú váhu.	Vzťah RDT-dokument má zmenenú váhu.	
3	Na obrazovke detailu repozitára v časti Assigned metadata klikneme na aktívny odkaz Publish.	Zmení sa verzia väzby Variant – Repozitár, odkaz Publish sa deaktivuje a has_changed sa zmení na false.	Zmenila sa verzia väzby Variant – Repozitár, odkaz Publish sa deaktivoval a has_changed sa zmenilo na false.	

3.2 Rozšírenie modelu používateľa

Aby som mohol pristupovať k funkciám systému, musím sa ako používateľ prihlásiť pomocou prihlasovacích údajov, ktoré sú uložené na externom serveri. Súčasne chcem, aby ma systém odlišoval od iných používateľov.

3.2.1 Analýza

Pre potreby systému je nutné umožniť prihlasovanie používateľov pomocou údajov uložených v rámci akademického informačného systému. Server umožňuje prihlasovanie pomocou LDAP¹. Z tohto dôvodu sa musí rozšíriť model používateľa aby umožňoval prihlasovanie takéhoto typu.

Na základe tejto požiadavky vzniká aj nutnosť rozlišovania pojmov používateľ a anotátor, pretože autorom anotácie už nebude len používateľ prihlásený v systéme.

Osobné údaje používateľa použité na jeho identifikáciu musia byť naďalej ochránené proti zneužitiu.

3.2.2 Návrh

Pred používaním systému sa používateľ prihlási pomocou svojho mena a hesla z univerzitného informačného systému. Odhlásiť sa môže v ľubovoľnom momente. Používateľ zostane prihlásený, pokiaľ sa neodhlási, alebo nevypne prehliadač. Po odhlásení bude používateľ navrátený na úvodnú obrazovku.

Model používateľa sa pozmení tak, že sa odstráni doterajšia funkcionálna prihlasovania a nahradí sa prihlasovaním pomocou externého servera. Model používateľa musí podporovať autentifikáciu pomocou LDAP. Po úspešnom prihlásení sa meno a heslo používateľa uloží do lokálnej databázy systému, aby mohli byť použité na autentifikáciu v prípade výpadku autentifikačného servera.

Do systému sa pridá model anotátora, aby sa odlišil od používateľa. Anotátor slúži na identifikovanie autora anotácie. Takýto autor nemusí byť používateľom systému a preto je potrebné odčleniť ho od používateľa. Anotátor môže mať viacero anotácií.

3.2.3 Implementácia

Používateľ je v systéme vytvorený pomocou gemu „Devise“. Je to rozšírený gem s dostatočnou funkcionálnosťou. Informácie o používateľovi uchováva v tabuľke „Users“, ktorá obsahuje viacero atribútov. Prihlasovanie pomocou LDAP implementujeme rozšírením gemu Devise pluginom „devise_ldap_authenticatable“ a „net_ldap“. Po úspešnom prihlásení sa používateľove údaje skopírujú do databázy. Používateľ je identifikovaný atribútom „login“ a heslom.

Prihlásenie je naďalej realizované pomocou formulára na uvítacej obrazovke. Obsahuje pole pre login a heslo akademického informačného systému. Prihlásenie sa vykoná stlačením tlačidla „Sign in“.

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.

¹ <http://tools.ietf.org/html/rfc4510>

Zmena prihlasovania modulom LDAP je zabezpečená vloženíím riadku „ldap_authenticatable“ do modelu používateľa. Výsledkom je, že prihlasovanie používateľa sa nevykoná lokálne pomocou databázy, ale pomocou autentifikačného servera.

3.2.4 Testovanie

Pre testovanie nebol vytvorený „Mock“ používateľ a neexistovala možnosť jeho vytvorenia na autentifikačnom serveri, preto bola funkcionálnosť otestovaná pomocou metódy čiernej skrinky.

Názov testu		Odhlásenie	ID	3-2-1
Rozhranie	Všetky obrazovky systému		autor	Martin Habdák
Účel	Odhlásenie používateľa			
Vstupné podmienky		Používateľ je prihlásený.		
Výstupné podmienky		Používateľ je odhlásený.		
Krok	Akcia	Očakávaná akcia	Skutočná akcia	
1	Klikneme na odkaz Sign out.	Odhlásenie používateľa, presmerovanie na privítaciu obrazovku.	Odhlásenie používateľa, presmerované na privítaciu obrazovku.	

Názov testu		Prihlásenie	ID	3-2-2
Rozhranie	Úvodná obrazovka systému		autor	Martin Habdák
Účel	Odhlásenie používateľa			
Vstupné podmienky		Používateľ je odhlásený.		
Výstupné podmienky		Používateľ je prihlásený.		
Krok	Akcia	Očakávaná akcia	Skutočná akcia	
1	Vyplníme prihlasovacie údaje a odošleme formulár .	Prihlásenie používateľa, zobrazenie úvodnej obrazovky.	Prihlásenie používateľa, zobrazila sa úvodná obrazovka.	

3.3 Výber vzťahov medzi RDT podľa vybraného RDT (obrazovka V1)

V obrazovke V1 (Príloha A: Obr. 13) po kliknutí na term zo zoznamu RDT by sa v pravej časti v tabuľke vzťahov medzi RDT mali vyfiltrovať iba vzťahy, v ktorých vystupuje označený term.

3.3.1 Analýza

Používateľ by mal po označení termu vidieť v tabuľke vzťahov vyfiltrované len vzťahy súvisiace s termom a po jeho odznačení sa by sa v tabuľke mali zobrazíť opäť všetky vzťahy. Táto funkcionálna uľahčí a sprehládní používateľovi prácu s množstvom vzťahov medzi termami v metadátach.

3.3.2 Návrh

Názvy termov v zozname RDT budú odkazy. Po kliknutí na term sa zavolá JavaScript funkcia pre vyfiltrovanie tých vzťahov, v ktorých vystupuje term na pozícii „From“ alebo „To“. Vybratý term bude vizuálne zvýraznený.

3.3.3 Implementácia

Názvy termov v zozname termov boli zmenené na odkazy (HTML <a> tagom). Po kliknutí na odkazy sa zavolá JavaScript funkcia pre vyfiltrovanie tabuľky v pravej časti obrazovky. Funkcia filtrovania je súčasťou pluginu Datatables, ktorý je použitý pre zobrazovanie tabuliek v systéme. Filtruje sa výskyt názvu termu v stĺpcoch „From“ a „To“. Vybratý term je vizuálne zvýraznený pomocou CSS. Po jeho odznačení (opätovnom kliknutí na term) zmizne zvýraznenie termu a v tabuľke vzťahov sa opäť zobrazia všetky vzťahy medzi termami v upravovaných metadátach.

3.3.4 Testovanie

Funkcionálnu overujú nasledovné akceptačné testy:

Názov testu		Vyfiltrovanie vzťahov podľa vybraného termu	ID	3-3-1
Rozhranie	Obrazovka V1, tabuľka so zoznamom RDT		autor	Petra Vrablecová
Účel	Overenie funkcionality filtrovania pre uľahčenie práce s metadátami			
Vstupné podmienky	Existujú metadáta s aspoň 3 termami a existujú vzťahy medzi termami.			
Výstupné podmienky	V tabuľke vzťahov vidíme vyfiltrované vzťahy podľa vybraného termu.			
Krok	Akcia	Očakávaná akcia	Skutočná akcia	
1	Klikneme na jeden z termov zo zoznamu termov.	Term je zvýraznený. V tabuľke vzťahov vidno len vzťahy súvisiace s termom.	Term bol zvýraznený. V tabuľke vzťahov bolo vidno len vzťahy, v ktorých vystupoval označený term.	

2	Klikneme na označený term.	Term stratí zvýraznenie. Tabuľka vzťahov bude obsahovať všetky vzťahy z variantu metadát.	Term nebol zvýraznený. V tabuľke vzťahov sa zobrazili všetky vzťahy variantu metadát.
---	----------------------------	---	---

3.4 Tabuľka susedov vrcholu nad grafom

V obrazovke V1 (Príloha A: Obr. 14) bolo potrebné upraviť tabuľku susedov uzla grafu tak, aby sa zobrazovala nad plochou grafu a nie vedľa grafu.

3.4.1 Analýza

Aby používateľ mal lepšie možnosti pracovať s grafom, bude tabuľka susedov označeného uzla zobrazená v hornom rohu nad plochou grafu a nie vedľa grafu, aby plocha grafu mohla byť väčšia.

3.4.2 Návrh

Pre dosiahnutie zväčšenia plochy grafu bude potrebné upraviť CSS pre zobrazované HTML elementy.

3.4.3 Implementácia

Bolo upravené CSS pre elementy `<div>` plochy grafu a tabuľky susedov. Ich prekryvanie bolo dosiahnuté pomocou úpravy CSS atribútov `position`.

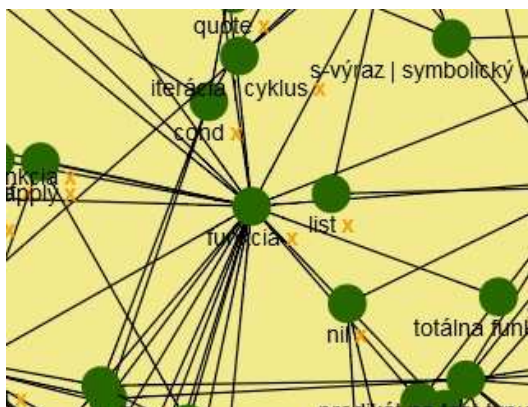
3.4.4 Testovanie

Funkcionalitu overujú nasledovné akceptačné testy:

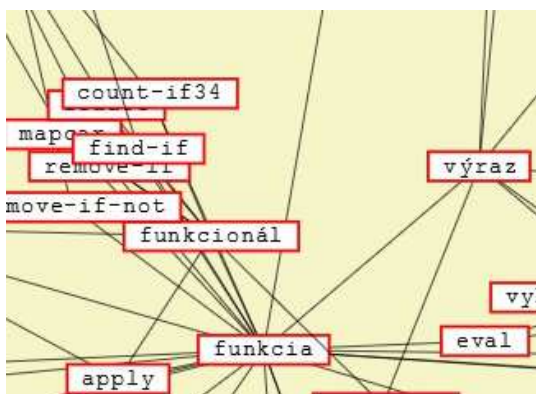
Názov testu		Zobrazenie tabuľky susedov.	ID	3-4-1
Rozhranie	Obrazovka V1, graf		autor	Petra Vrablecová
Účel	Overenie správneho zobrazenia tabuľky susedov			
Vstupné podmienky	Existujú metadáta s termami a existujú vzťahy medzi termami.			
Výstupné podmienky	Tabuľka susedov je zobrazená nad plochou grafu.			
Krok	Akcia	Očakávaná akcia	Skutočná akcia	
1	Zobrazíme metadáta vo fomre grafu a klikneme na jeden uzol.	Tabuľka susedov je zobrazená nad plochou grafu a obsahuje zoznam susedov označeného uzla.	Tabuľka bola zobrazená nad plochou grafu a obsahovala zoznam susedov označeného uzla.	

3.5 Vizualizácia uzlov ako textových

Zmena zobrazovania preddefinovaných vrcholov na prívetivejšie zobrazenie.



Obr. 1. Pred úpravami.



Obr. 2. Po úpravách.

3.5.1 Implementácia

V `graph_view.js` je pri inicializácii (funkcia `init(rdts, relationships, relationshipTypes)`) vytvorený nový typ vrcholu. Typ je implementovaný v rámci funkcie knižnice Jit `jit.ForceDirected.Plot.NodeTypes.implement()`. Tá ako parametre berie štruktúru skladajúcu sa z nových typov vrcholov. Pre každý z týchto typov museli byť implementované funkcie `render` a `contains`. Funkcia `render` má na starosti vykresľovanie nových uzlov. To sme riešili použitím HTML5 prvku `canvas`. Funkcia `contains` zisťuje, či sa hľadáček myši nachádza akurát nad aktuálnym vrcholom. Obe funkcie museli byť napísané úplne nanovo, aby sa podarilo dosiahnuť požadovaný cieľ.

3.5.2 Testovanie

Názov testu		Test ohraničenia vrcholov	ID	3-5-1
Rozhranie	Okno grafu V1		autor	Martin Gajdoš
Účel	Otestovanie správneho zobrazenia			
Vstupné podmienky		Načítaný graf.		
Výstupné podmienky		Žiadne.		
Krok	Akcia	Očakávaná akcia	Skutočná akcia	
1	Náhľad na graf.	Správne zobrazené vrcholy.	Správne zobrazené vrcholy.	
2	Prejdenie nad vrchol myškou.	Zmena kurzora na štvorhrannú šípku.	Zmena kurzora na štvorhrannú šípku.	
3	Pohnutie s uzlom metódou drag and drop.	Zmena pozície uzlu.	Zmena pozície uzlu.	

3.6 Filtrovanie tabuliek na úrovni stĺpcov

3.6.1 Analýza

V časti analýzy sme sa pokúsili zistiť, ktorá z hľadaných metód v plugine datables bude pre túto úlohu najvhodnejšia. Metódy boli nájdené dve, pričom prvá z nich bola jednoduchšia a druhá vyžadovala prechod do vyššej verzie datables. Po nasadení a testovaní prvej metódy vyšlo najavo, že je treba prejsť na druhú a to z dôvodu rôznorodého zobrazovania vo viacerých prehliadačoch.

3.6.2 Návrh

Pod hlavičkou tabuliek, sa mali zobrazovať vstupné polia pre zadanie hodnoty filtrovania, pričom týchto polí malo byť toľko koľko je stĺpcov v danej tabuľke. Užívateľ si vyberie stĺpec, ktorý chce filtrovať tým že klikne na konkrétne textové pole v stĺpci, ktorý chce filtrovať a zadá hodnotu. Vyfiltrovaná tabuľka sa hneď po zadaní každého písmena zobrazí namiesto bežnej tabuľky.

3.6.3 Implementácia

Implementácia prebiehala v jazyku JavaScript, v ktorom je Datables plugin vytvorený, pričom tento kód bol po implementácii Coffee Scriptu prepísaný. Kód bol implementovaný do všetkých filtrovateľných tabuliek ako sú repozitáre, varianty, dokumenty atd..

4 Šprint č. 7 – Grieg

4.1 Preview dokumentu pri akcii hover v obrazovke RDV1

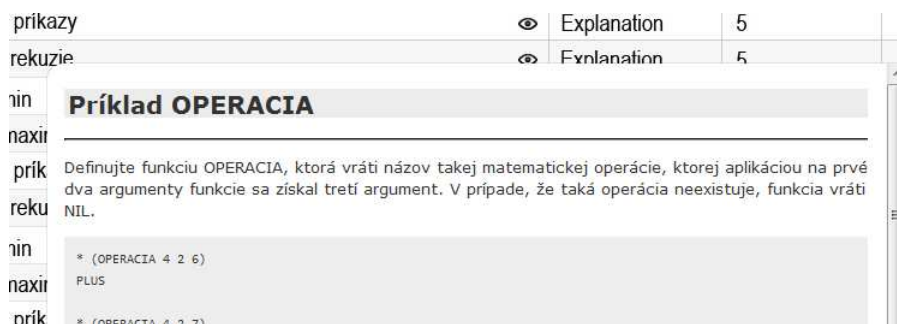
Ako používateľ systému chcem pri editovaní vzťahov k dokumentu nahliadnuť do jeho obsahu bez nutnosti presunutia na inú obrazovku.

4.1.1 Analýza

Pri editácii vzťahov k dokumentu v obrazovke RDV1 (Príloha A: Obr. 15) používateľ často potrebuje nahliadnuť do obsahu dokumentu aby sa vedel lepšie orientovať pri práci s ním. Nahliadnutie je mnohokrát nutné len na krátku dobu alebo len na prezretie malej časti textu. Prechádzanie na inú obrazovku by v tomto prípade nebolo vhodné a spomaľovalo by používateľa. Navyše by mohol stratiť neuložené zmeny. Zobrazený obsah dokumentu by mal byť ľahko čitateľný a štruktúrovaný podľa skutočného vzhľadu dokumentu.

4.1.2 Návrh

Aby používateľ mohol vidieť obsah dokumentu bez nutnosti prepínania obrazovky, pridáme pre každý z nich do obrazovky RDV1 ikonu znázorňujúcu oko. Ikona bude spúšťať funkcionality takzvaného „Nahliadnutia“. Spočíva v zobrazení dokumentu podobným spôsobom, akým sa zobrazuje „tooltip“. Príklad takéhoto zobrazenia vidieť na obr. 3. Používateľ musí kurzor ponechať nad ikonou oka po dobu približne 0,5 sekundy. Až po uplynutí tejto doby sa zobrazí tooltip obsahujúci dokument vo formáte HTML. Do tohto formátu sa asynchrónne prevedie pomocou transformácie z XML.



Obr. 3. Náhľad dokumentu pomocou ikony oka.

4.1.3 Implementácia

Náhľad sa musí realizovať asynchrónne, preto využijeme pri jeho implementácii technológiu AJAX. Pri prejení nad myšou sa automaticky spustí získanie obsahu dokumentu vo formáte HTML. Do tohto formátu sa dokument prevedie pomocou XSLT transformácie. Ak používateľ opustí ikonu do 0,5 sekundy, akcia sa ihneď preruší a nedôjde k zobrazeniu. Dôvod prečo sa transformácia spustí hneď po prejení kurzorom nad ikonou oka a nie až po 0,5 sekunde je ten, že transformácia môže trvať viac ako uvedený čas. Okamžitým spustením sa teda ušetrí pol sekundy a používateľské rozhranie je tak responzívnejšie.

Samotný tooltip je implementovaný pomocou elementu <div>, ktorý je na začiatku neviditeľný. Pri spustení zobrazenia sa vždy naplní jeho obsah pretransformovaným dokumentom a následne sa zobrazí na pozícii kurzora. Po opustení náhľadu kurzorom sa tento náhľad skryje.

4.1.4 Testovanie

Názov testu	Zobrazenie náhľadu dokumentu	ID	4-1-1
Rozhranie	Obrazovka RDV1	autor	Martin Habdák
Účel	Zobrazenie náhľadu dokumentu pri akcii hover nad ikonou oka		
Vstupné podmienky	Používateľ je prihlásený, obsah dokumentu je validný voči schéme.		
Výstupné podmienky	Používateľ je prihlásený, zobrazil sa náhľad dokumentu.		
Krok	Akcia	Očakávaná akcia	Skutočná akcia
1	Prenesieme kurzor nad ikonu oka a ponecháme ho na tomto mieste aspoň 0,5 sekundy.	Zobrazenie náhľadu obsahu dokumentu.	Zobrazil sa náhľad obsahu dokumentu.

Názov testu	Zobrazenie chybovej hlášky náhľadu	ID	4-1-2
Rozhranie	Obrazovka RDV1	autor	Martin Habdák
Účel	Zobrazenie chybovej hlášky pri náhľade na dokument		
Vstupné podmienky	Používateľ je prihlásený, obsah dokumentu nie je validný voči schéme.		
Výstupné podmienky	Používateľ je prihlásený, zobrazila sa chybová hláška.		
Krok	Akcia	Očakávaná akcia	Skutočná akcia
1	Prenesieme kurzor nad ikonu oka a ponecháme ho na tomto mieste aspoň 0,5 sekundy.	Zobrazenie chybovej hlášky dokumentu.	Zobrazila sa chybová hláška.

Názov testu	Nezobrazenie náhľadu dokumentu		ID	4-1-3
Rozhranie	Obrazovka RDV1		autor	Martin Habdák
Účel	Nezobrazenie náhľadu dokumentu a chybovej hlášky			
Vstupné podmienky	Používateľ je prihlásený.			
Výstupné podmienky	Používateľ je prihlásený.			
Krok	Akcia	Očakávaná akcia	Skutočná akcia	
1	Prenesieme kurzor nad ikonu oka a neponecháme ho na tomto mieste aspoň 0,5 sekundy.	Nezobrazenie náhľadu dokumentu a chybovej hlášky.	Nezobrazil sa náhľad dokumentu, ani chybová hláška.	

4.2 "Advanced" tab na zobrazenie anotácií

4.2.1 Analýza

Na vytvorenie tejto úlohy sa vyskytli dve odlišné alternatívy riešenia. Prvá alternatíva pojednáva o tom, urobiť túto pomocou pluginu Datables s tým, že je k tomu potrebná dokumentácia aj príklady kódu. Druhá alternatíva bola vyriešiť túto úlohu pomocou Coffee Scriptu. Aj keď prvé riešenie sa zdalo byť rýchlejšie nepokrývalo by všetky potreby pre zobrazovanie a prepočítavanie okolia. Preto bola zvolená alternatíva číslo dva.

4.2.2 Návrh

Pokročilé zobrazovanie okolia bolo navrhnuté tak, aby sa okolie zobrazovalo pod daným riadkom tabuľky, pričom v tomto riadku by bol vymedzený stĺpec pre tlačidlo zobrazenia. Zobrazovanie by malo dve roviny a to pri riadku a hromadné zobrazovanie všetkých anotácií v tabuľke pomocou tlačidla pod tabuľkou. Dizajn bol navrhnutý tak, aby zapadol do rozhrania Datables.

4.2.3 Implementácia

Implementovanie prebiehalo tak pomocou jazyka Coffee Script. Podarilo sa implementovať presne podľa stanoveného návrhu, pričom samotná funkcionálna bola implementovaná len do tabuľky prislúchajúcich anotácií k dokumentu.

4.3 Pop-up v grafe

Jednotlivé vrcholy v grafe je možné upraviť pravým kliknutím myši na požadovaný vrchol.

4.3.1 Použitie

Používateľ stlačí pravé tlačidlo myši nad vrcholom v grafovom zobrazení. Poskytnú sa mu možnosti, čo si s vrcholom praje spraviť. Zvolí si jednu z možností a vykoná príslušné akcie.

4.3.2 Implementácia

Pri vytváraní nového grafu *ForceDirected* boli ku nemu pridané udalosti, ktoré reagujú na akcie myši. Ak je akcia pravého kliknutia vykonaná nad vrcholom, zobrazí sa okno v mieste kliknutia, ktoré poskytuje možnosti: zmazania uzlu, úpravy uzlu, zobrazenia susedov v tabuľke, pridanie nového prepojenia k ďalšiemu uzlu. Pre každú z týchto akcií sa opätovne vytvorí nové dialógové okno s požadovanou funkcionalitou.

4.3.3 Testovanie

Názov testu		Test zobrazenie pop-up menu	ID	4-3-3
Rozhranie	Okno grafu V1		autor	Martin Gajdoš
Účel	Otestovanie správneho zobrazenia pop-up menu			
Vstupné podmienky	Načítaný graf.			
Výstupné podmienky	Žiadne.			
Krok	Akcia	Očakávaná akcia	Skutočná akcia	
1	Stlačenie pravého tlačítka nad vrcholom.	Zobrazenie pop-up menu.	Zobrazenie pop-up menu.	
2	Výber editácie vrcholu.	Zobrazené okno editácie.	Zobrazené okno editácie.	
3	Výber vymazania vrcholu.	Vrchol vymazaný z grafu.	Vrchol vymazaný z grafu.	
4	Výber zobrazenia prislúchajúcich vzťahov.	Zobrazené vzťahy.	Zobrazené vzťahy.	
5	Výber pridania nového vzťahu.	Zobrazené okno pridania vzťahu.	Zobrazené okno pridania vzťahu.	

4.4 Graf property panel

V zobrazení grafu na obrazovke V1 (Príloha A: Obr. 14) bude možné meniť vlastnosti zobrazenia grafu. Bude možné ohraničiť hĺbku zobrazovaných susedov zvoleného uzla, zvoliť typy zobrazovaných vzťahov medzi uzlami, typy zobrazenia hrán medzi uzlami a meniť spôsob rozloženia uzlov v grafe.

4.4.1 Analýza

Používateľ pre pohodlnejšiu prácu s grafom reprezentujúcim variant metadát potrebuje nastavenia, pomocou ktorých si môže upraviť jeho zobrazenie. Preto je potrebné mu sprístupniť v grafe panel s takýmito nastaveniami. Aby panel nezaberal zbytočne plochu grafu, bude ho možné skryť. Zmeny v nastaveniach by mali byť ihneď viditeľné na grafe.

4.4.2 Návrh

Panel s nastaveniami sa bude nachádzať v dolnej časti grafu. Ovládacie prvky jednotlivých nastavení budú zvolené podľa návrhu obrazovky. Pri interakcii s nastaveniami sa budú spúšťať JavaScript funkcie využívajúce funkcionality knižnice JIT, v ktorej je implementovaný graf.

4.4.3 Implementácia

Panel s nastaveniami a jeho ovládacie prvky boli implementované pomocou HTML a knižnice JQuery. Funkcie pre ohraničovanie susedov, zobrazovanie typov hrán, filtrovanie typov vzťahov a menenie rozloženia uzlov grafu boli implementované pomocou knižnice JIT, ktorá bola použitá na implementáciu grafu.

4.4.4 Testovanie

Funkcionalitu overujú nasledovné akceptačné testy:

Názov testu		Menenie nastavení zobrazenia grafu	ID	4-4-1
Rozhranie	Obrazovka V1, graf		autor	Petra Vrablecová
Účel	Overenie funkcionality panelu s nastaveniami zobrazenia grafu			
Vstupné podmienky	Existujú metadáta s termami a existujú vzťahy rôznych typov medzi termami.			
Výstupné podmienky	Zobrazenie grafu je prispôbené zvoleným nastaveniam v paneli.			
Krok	Akcia	Očakávaná akcia	Skutočná akcia	
1	Klikneme na tlačidlo Options v prvom dolnom rohu grafu.	Vysunie sa panel s možnosťami zobrazenia grafu.	Vysunul sa panel s možnosťami nastavenia zobrazenia grafu.	
2	Zvolíme uzol, v grafe a nastavíme na slideri hĺbku zobrazovaných susedov uzla.	Uzly nachádzajúce sa vo väčšej hĺbke ako je tá zvolená na slideri, zmiznú.	Uzly vo väčšej hĺbke zmizli.	
3	Zvolíme zobrazenie hrán grafu podľa váhy vzťahov.	Hrany medzi uzlami sa stenčia/zhrubnú podľa váhy vzťahu, ktorý reprezentujú.	Hrany sa stenčili/zhrubli podľa váhy vzťahu, ktorý reprezentujú.	
4	Zvolíme rozloženie uzlov grafu a stlačíme tlačidlo Reset.	Uzly v grafe sa preusporiadajú podľa zvoleného rozloženia.	Uzly v grafe sa preusporiadali podľa zvoleného rozloženia.	
5	Obmedzíme zobrazovanie na jeden typ vzťahu.	Vzťahy ostatných typov zmiznú.	Vzťahy ostatných typov zmizli.	

4.5 Komunikácia systému s klientmi – Obsah

Cieľom je oboznámiť klientov systému o zmenách v obsahu a anotáciách.

4.5.1 Analýza

Zmeny v obsahu repozitárov, resp. kurzov (v systéme ALEF), bude možné realizovať už len prostredníctvom systému FELA, čím sa redukuje komunikácia o zmenách v obsahu dokumentov len na požiadavku ALEFu a následnú odpoveď FELA so zmeneným obsahom.

- I. Opis: Vieme, že FELA obsahuje novú verziu obsahu pre setup a chceme ho načítať do ALEFu.
Akcia:
 1. Stlačíme button `↑ Update ↓` pre získanie verzie obsahu.
 2. Kurz v ALEFe sa uzamkne.
 3. ALEF pošle požiadavku na obnovu dát, FELA pošle odpoveď.
 4. Ak `ALEF.verzia != FELA.verzia`, aktualizuje sa stav v ALEFe, nepokračuje sa ďalej, čaká sa.
 5. FELA pošle obsah, ktorým sa nahradí obsah v danom setupe v ALEFe.
 6. Uvoľní sa zámok z kurzu v ALEFe (z každého dokumentu).

- II. Opis: Chceme upraviť obsah vo FELA (nový repozitár, veľké zmeny v celom repozitári).
Akcia:
 1. Prihlásim sa do FELA a robím zmeny.
 2. Po dokončení zmien kliknem na `↑ Publish ↓`, čím sa verzia repozitára inkrementuje a stane sa dostupnou pre ALEF.

1.1.5 Návrh

Pre repozitáre bude zavedený podobný systém zachytenia zmien, ako bol implementovaný pre metadáta. Úpravami v dokumentoch sa nastaví príznak `has_changed` v modeli repozitára, čím sa umožní spustenie akcie Publish, ktorá zvyšuje verziu repozitára.

Obsah bude možné získať v archíve, ktorý sa používa pre export.

1.1.6 Implementácia

Zaznamenanie zmeny v repozitári je zabezpečené Callback funkciou v modeli dokumentu. Akcia `publish` v controlleri `RepositoriesController` zmení príznak `has_changed` na `false` a zvýši číslo verzie repozitára o 1.

Pre získavanie obsahu repozitára sa používajú metódy modelu `Repository`, ktoré zabezpečujú export repozitára.

4.5.2 Testovanie

Názov testu		Zmena verzie obsahu repozitára	ID	4-5-1
Rozhranie	Dokument - Edit		autor	Martin Franta
Účel	Oboznámenie klientov o zmenách v obsahu			
Vstupné podmienky	Repozitár obsahuje dokument, jeho príznak has_changed je false.			
Výstupné podmienky	Verzia repozitára je zmenená, jeho príznak has_changed je false.			
Krok	Akcia	Očakávaná akcia	Skutočná akcia	
1	Vo formulári úpravy dokumentu upravíme dokument a uložíme.	Obsah dokumentu je zmenený.	Obsah dokumentu sa zmenil.	
2	Na úvodnej obrazovke v časti Repositories klikneme na aktívny odkaz Publish pre zvolený repozitár.	Odkaz Publish sa deaktivuje, verzia sa zvýši, has_changed zmení hodnotu na false.	Odkaz Publish sa deaktivoval, verzia sa zvýšila, has_changed sa zmenilo hodnotu na false.	

5 Šprint č. 8 – Haydn

5.1 Komunikácia systému s klientmi – Anotácie

Cieľom je zabezpečenie komunikácie ohľadom anotácií medzi systémom a klientom.

5.1.1 Analýza

Z pohľadu anotácií boli určené tri dôležité prípady použitia:

- △ Kontrola aktuálnej verzie obsahu

Z ALEFu sa posielajú anotácie, ktoré sa od poslednej kontroly zmenili.

- △ Aktualizácia obsahu z úložiska

V tomto bode bude doplnený export repozitára o anotácie, ktoré prislúchajú k dokumentom v repozitári.

- △ Editácia obsahu zvoleného dokumentu

Používateľ sa môže priamo z ALEFu prepnúť do FELA, kde môže upraviť obsah zvoleného dokumentu. Posielajú sa pritom (zmenené a nové) anotácie naviazané len na zvolený dokument.

V tejto fáze bolo potrebné vyriešiť tiež oprávnenia k prístupu k jednotlivým akciám systému FELA. Pre komunikáciu ohľadne obsahu je možné používať tajné API kľúče, ktoré sa generujú pri vytvorení repozitára.

5.1.2 Návrh

Pre odosielanie anotácií bol zvolený formát JSON v tvare:

```
[
  {
    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'
      },
      ...
    ]
  },
  ...
],
```

...

]

V tejto fáze vývoja budú vytvorené akcie, ktoré dokážu informácie v takomto formáte spracovať a následne upraviť existujúce anotácie, prípadne vytvoriť nové. JSON odpoveď získa systém HTTP požiadavkou svojmu klientovi.

Doplnené anotácie v ZIP archíve budú v tvare XML súboru vo formáte:

```
<?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>
```

Pri akcii editácie dokumentu, ktorá bola vyvolaná zo strany klienta, sa posielajú anotácie vo forme JSON požiadavky:

```
[
  {
    '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'
  },
  ...
]
```

Spracovanie anotácií bude prebiehať podobne ako v prípade anotácií pre celý repozitár. Po spracovaní bude používateľ presmerovaný na formulár úpravy dokumentu.

5.1.3 Implementácia

Do modelu Annotation boli pridané metódy, ktoré z asociatívneho poľa (dátová štruktúra hash), vytvárajú inštancie modelu Annotation. Potom na základe existencie anotácie v systéme FELA, model uložia ako nový, alebo nahradia už existujúcu anotáciu.

Metóda `get_annotations` získava odpoveď zo zadanej URL adresy, kým vyššie uvedené metódy ju spracujú.

Metóda `create_annotation_xml` v modeli Repository zabezpečuje vytvorenie XML súboru podľa špecifikovaného formátu.

5.1.4 Testovanie

Testovanie prebiehalo jednotkovými testami implementovaných metód a postupov, nakoľko prepojenie FELA s klientom (ALEF) v ôsmom šprinte ešte nebolo zrealizované.

Testy sú obsiahnuté v Rspec súboroch `models/annotation_spec.rb` a `models/repository_spec.rb`.

5.2 Edit anotácie

Anotáciu by malo byť možné editovať pri zobrazení dokumentu po kliknutí na anotačný pásik alebo prostredníctvom tabuľky s anotáciami zobrazenej vedľa dokumentu.

5.2.1 Analýza

Používateľ potrebuje v obrazovke DV1 (Príloha A: Obr. 17) editovať anotácie prívetivo, t.j. tak, aby pri editovaní nezadával začiatočnú a koncovú pozíciu anotácie v texte, ale mohol ju priamo označiť. Tento spôsob editovania anotácii korešponduje s novým modelom anotácie, ktorý obsahuje anotovaný text a jeho okolie namiesto číselných pozícií anotovaného textu.

5.2.2 Návrh

Po kliknutí na anotáciu v anotačnom pásiku alebo kliknutie na editovanie anotácie v tabuľke anotácií vedľa dokumentu bude mať používateľ možnosť znova označiť text v dokumente a nastaviť ho ako text anotácie.

5.2.3 Implementácia

Editovanie anotácie je implementované ako dialóg pomocou HTML a knižnice JQuery. Dialóg obsahuje prvky pre zmenu atribútov anotácie – obsah, typ, text anotácie a tlačidlo na aktualizovanie textu anotácie na aktuálne označený text v dokumente. Po stlačení tlačidla Update sú informácie z dialógu odoslané pomocou technológie AJAX do databázy.

5.2.4 Testovanie

Funkcionalitu overujú nasledovné akceptačné testy:

Názov testu		Edit anotácie	ID	5-2-1
Rozhranie	Obrazovka DV1		autor	Petra Vrablecová
Účel	Overenie nového spôsobu editovania anotácie.			
Vstupné podmienky	Je vytvorený dokument s aspoň jednou anotáciou.			
Výstupné podmienky	Anotácia je aktualizovaná.			
Krok	Akcia	Očakávaná akcia	Skutočná akcia	
1	Klikneme na anotáciu v anotačnom pásiku.	Otvorí sa dialóg pre editovanie anotácie.	Otvoril sa dialóg pre editovanie anotácie.	
2	Označíme text v dokumente a klikneme na tlačidlo Change to selected v dialógu editovania anotácie.	Text anotácie v dialógu sa zmení na označený text.	Text v dialógu anotácie sa zmenil na označený text v dokumente.	
3	Klikneme na tlačidlo Update.	Dialóg sa zatvorí. Informácie v databáze sa aktualizujú a anotácia v anotačnom pásiku sa zobrazí na správnom mieste.	Dialóg sa zatvoril, anotácia v pásiku sa aktualizovala.	

5.3 Vkladanie anotácií podľa ALEFu

Tvorba novej anotácie inšpirovaná systémom ALEF.

5.3.1 Použitie

V náhľade na dokument používateľ označí myšou text, ktorý si praje anotovať. Nad ním sa mu zobrazia možnosti, čo s vybraným textom vykonať. Zatiaľ bola implementovaná funkcia vytvorenia novej anotácie.

5.3.2 Implementácia

K HTML elementu v ktorom sa náhľad na dokument nachádza boli pridané nové akcie pri vzniknutých udalostiach stlačenia a pustení myši. Pri stlačení sa zruší staré označenie. Pri pustení myši je staré je okolo nového výberu vytvorené nové označenie vo formáte HTML elementu ktorý je generovaný knižnicou Rangy. Nad ním sú zobrazené možnosti.

5.3.3 Testovanie

Názov testu		Test funkcionality pridania anotácie	ID	5-3-1
Rozhranie	Okno detailu dokumentu DE1		autor	Martin Gajdoš
Účel	Otestovanie funkcionality			
Vstupné podmienky	Načítaný validný dokument.			
Výstupné podmienky	Žiadne.			
Krok	Akcia	Očakávaná akcia	Skutočná akcia	
1	Označíme text v dokumente.	Špeciálne zvýraznenie textu a objavenie možnosti New nad označením.	Špeciálne zvýraznenie textu a objavenie možnosti New nad označením.	
2	Výber možnosti New.	Špeciálne zvýraznenie textu a objavenie možnosti New nad označením.	Špeciálne zvýraznenie textu a objavenie možnosti New nad označením.	
3	Výber možnosti New.	Zobrazenie okna novej anotácie.	Zobrazenie okna novej anotácie.	

5.4 Import repozitára + metadát: infobox s výberom

Ako používateľ potrebujem pri importovaní súborov vybrať tie, ktoré chcem importovať. Súčasne sa chcem rozhodnúť, či budem obsah súborov validovať voči schéme.

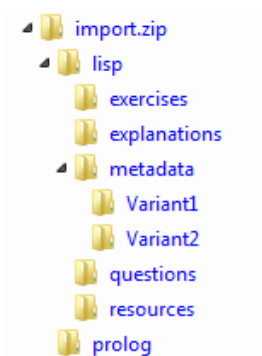
5.4.1 Analýza

Pri importovaní používateľ nemusí vždy chcieť importovať všetky súbory obsiahnuté v archíve. Je to z dôvodu, že by import mohol trvať veľmi dlhú dobu, nechce zmeniť už existujúce súbory alebo nepotrebuje importovať všetok obsah. Počas importu je nutné aj validovať súbory voči schéme. Táto validácia by mala byť tiež spustená na požiadanie používateľa. Pri vzniku konfliktov treba evidovať ich počet.

5.4.2 Návrh

Po vybraní archívu používateľom sa mu zobrazí dialóg, v ktorom je znázornená štruktúra archívu. Vyberie si požadované súbory, ktoré chce importovať a zvolí si, či chce vykonať ich validáciu. Následne voľbu potvrdí a spustí sa samotný import súborov. Po skončení sa vypíše hláška o úspechu importu.

Pri importovaní sa budú evidovať konflikty a ich počet ukladať do databázy. Počet konfliktov bude tvoriť základ pre model histórie importov. Ten sa neskôr bude rozširovať podľa potrieb zákazníka.



Obr. 4. Štruktúra archívu.

Pre potreby importu sa mierne upraví štruktúra archívu. Archív s údajmi bude obsahovať jeden alebo viac kurzov. Každý kurz predstavuje jeden adresár v archíve. V adresári pre kurz je štruktúra podobná ako doteraz. Obsahuje adresár pre každý typ dokumentu (exercises, explanations, questions). Ďalej obsahuje adresár „resources“ kde sa nachádzajú obrázky k dokumentom. Súbor manifest.MF nebude už potrebný. Štruktúra adresára „metada“ sa trochu zmení. Pre každý variant bude obsahovať jeden alebo viac adresárov (Variant1, Variant2 ...). V adresári pre daný variant budú umiestnené RDT a vzťahy tak, ako boli doteraz v adresári metadata. Názov variantu sa bude čerpať z názvu adresára, v ktorom sa metadáta nachádzajú. Štruktúru archívu možno vidieť na obr. 4.

5.4.3 Implementácia

Výber možností importu sa vykonáva pomocou dialógového okna. Toto okno sa zobrazí po výbere archívu. Je inicializované z daného elementu <div>, ktorý sa pred zobrazením naplní obsahom podľa archívu. 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 spustí importovanie asynchrónne. Asynchrónnosť zaisťuje spustenie akcie importovania v samostatnom procese.

História importov sa implementuje modelom, kde hlavným atribútom je čas vytvorenia záznamu o importe. Model obsahuje aj atribút poslednej zmeny, podľa ktorého sa dá vypočítať trvanie importu.

5.4.4 Testovanie

Názov testu		Importovanie s validáciami	ID	5-4-1
Rozhranie	Úvodná stránka (obrazovka Home)		autor	Martin Habdák
Účel	Overovanie integrity importovaných súborov podľa schémy			
Vstupné podmienky	Používateľ je prihlásený, používateľ zaškrtnol možnosť validácie súborov.			
Výstupné podmienky	Používateľ je prihlásený, súbory sú validované voči schéme.			
Krok	Akcia	Očakávaná akcia	Skutočná akcia	
1	Na hlavnej obrazovke zvolíme akciu importovania.	Zobrazenie výberu súboru.	Zobrazil sa výberu súboru.	
2	Vyberieme požadovaný archív a potvrdíme.	Zobrazenie okna s možnosťami importu.	Zobrazilo sa okno s možnosťami importu.	
3	Vyberieme možnosti validácie súborov a potvrdíme import.	Zmiznutie okna importu a zobrazenie hlášky po skončení importovania.	Zmizlo okno importu a zobrazila sa hláška po skončení importovania.	

Názov testu	Importovanie vybraných súborov	ID	5-4-2
Rozhranie	Úvodná stránka (obrazovka Home)	autor	Martin Habdák
Účel	Importovanie súborov, ktoré boli vybrané používateľom		
Vstupné podmienky	Používateľ je prihlásený, používateľ zaškrtnol súbory, ktoré majú byť importované.		
Výstupné podmienky	Používateľ je prihlásený, naimportované sú len súbory vybrané používateľom.		
Krok	Akcia	Očakávaná akcia	Skutočná akcia
1	Na hlavnej obrazovke zvolíme akciu importovania.	Zobrazenie výberu súboru.	Zobrazil sa výberu súboru.
2	Vyberieme požadovaný archív a potvrdíme.	Zobrazenie okna s možnosťami importu.	Zobrazilo sa okno s možnosťami importu.
3	Vyberieme súbory na importovanie a potvrdíme import.	Zmiznutie okna importu a zobrazenie hlášky po skončení importovania.	Zmizlo okno importu a zobrazila sa hláška po skončení importovania.

5.5 Zapamätanie si pozícií vrcholov v grafe

Graf nie je po každom zobrazení generovaný nanovo, ale pozície jednotlivých uzlov sú načítavané z databázy.

5.5.1 Použitie

Po vykonaní zmien pozícií v grafovom rozhraní zvolí používateľ možnosť *Apply changes*.

5.5.2 Implementácia

Za pomoci migrácií boli do tabuľky RDT pridané vlastnosti `posx` a `posy`, ktoré určujú pozície vrcholov pri načítaní grafu. Pri zaslaní požiadavky od klienta na otvorenie okna variantu je do odpovede vygenerovaná štruktúra prislúchajúcich RDT a ich vzťahov, vrátane všetkých vlastností, teda aj pozícií. Táto štruktúra je na strane klienta prepísaná do formátu JSON a poskytnutá knižnici Jit na vykreslenie grafu. Pozícia na ktorej sa má vrchol nachádzať je do grafu zadaná pri vytváraní vlastného typu uzlu podmienkou:

```
if (node.getData('f_x') != null && node.getData('f_y') != null) {
    node.endPos.setc(node.getData('f_x'), node.getData('f_y'));
}
```

5.5.3 Testovanie

Názov testu		Test zapamätanie vrcholov	ID	5-5-1
Rozhranie	Okno grafu V1		autor	Martin Gajdoš
Účel	Otestovanie funkcionality zapamätania pozícií			
Vstupné podmienky	Načítaný graf.			
Výstupné podmienky	Žiadne.			
Krok	Akcia	Očakávaná akcia	Skutočná akcia	
1	Zmeníme pozíciu troch vrcholov.	Tri vrcholy s novou pozíciou.	Tri vrcholy s novou pozíciou.	
2	Zvolíme možnosť Apply changes.	Potvrdenie o uložených zmenách.	Potvrdenie o uložených zmenách.	
3	Opätovne načítame graf.	Opätovne načítaný graf.	Opätovne načítaný graf.	
4	Overíme zapamätanie nových pozícií vrcholov.	Vrcholy sa nachádzajú na mieste, na ktorom boli pri ukladaní zmien.	Vrcholy sa nachádzajú na mieste, na ktorom boli pri ukladaní zmien.	

6 Opis prototypu

6.1 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.

6.1.1 Home

Úvodná obrazovka systému (Príloha A: Obr. 11) 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

Home 2

1 - Home Signed in as xgajdosm2. Sign out

Repositories

Search all columns:

Name	Version	hasChanged	Documents#	Actions			
<input type="text" value="Name"/>	<input type="text" value="Version"/>	<input type="text" value="hasChanged"/>	<input type="text" value="Documents#"/>				
lisp	0	true	297	Publish	S	E	X

3 4 5

Add Export selected Import: Vybrat súbor Nie je vybr...iadny súbor

Variants

Name	V.	Changed	#RDTs	#RDT relationships	#Repositories	Actions			
Mária_Bieliková	0	true	164	247	1	S	E	Publish	X
Novy	0	true	2	1	0	S	E	Publish	X

6

7

Add

Obr. 5. Obrazovka Home.

Legenda

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

6.1.2 Repo Detail (RD2)

Pri zobrazení konkrétneho repositára (Príloha A: Obr. 12) sa dostávame do detailu repositára. Táto časť systému obsahuje varianty metadát, ktoré sú priradené k zobrazenému repositáru a tiež dokumenty, ktoré tento repositár obsahuje.

Dokumenty

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

6.1.3 Metadata Detail – table (V1)

Obrazovka V1 (Príloha A: Obr. 13) 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 funkcionality:

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 polí 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.

Zoraďovanie a vyhľadavanie

Oba zoznamy majú možnosť zoraďovania 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.

6.1.4 Metadata Detail – graph (V1)

Táto obrazovka (Príloha A: Obr. 14) slúži na manipuláciu s metadátami vo forme grafu. Poskytuje väčšinu funkcionality predchádzajúcej obrazovky, ale pomocou iných ovládacích prvkov.

The screenshot shows the 'Metadata Detail - graph (V1)' interface. At the top, the user 'Mária Bieliková' is logged in. The interface is divided into several sections:

- Top Bar:** User name 'Mária Bieliková', 'Home', 'Signed in as xgajdosm2', and 'Sign out'.
- Left Sidebar (RDTs):** A table listing various RDTs with columns for 'Name' and 'Actions'. An 'Add' button is at the bottom.
- Main Graph Area:** A large network graph with nodes and edges. A 'Graph view' tab is active, and a 'Table' tab is also visible. An 'Apply changes' button is in the top right of the graph area.
- Right Sidebar (výraz):** A search box and a list of connections to the selected RDT.
- Bottom Control Panel:** Settings for 'Neighbourhood size', 'Layout' (String Layout), 'Relationship weight' (None), and 'Relationship filter' (is-a, prerequisite-to, related-to).

Obr. 6. 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

6.1.5 Repo Detail (RDV1)

Obrazovka RDV1 (Príloha A: Obr. 15) 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 preniesie používateľa na obrazovku DV1, ktorej funkcionálna je opísaná nižšie.

Zoraďovanie a vyhľadávanie

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

6.1.6 Document Detail – edit (DE1)

Obrazovka DE1 (Príloha A: Obr. 16) slúži na editovanie XML dokumentov.

Obr. 7. 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
7. Odkaz na obrazovku zobrazenia dokumentu
8. Vymazanie zvolených RDT

6.1.7 Document Detail – view (DV1)

Obrazovka DV1 (Príloha A: Obr. 17) slúži na prezeranie dokumentov, ale najmä na prácu s anotáciami.



Obr. 8. 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. Tlačidlo pre zobrazenie/skrytie príbuzných entít
5. Zobrazenie histórie dokumentu, možnosť návratu k staršej verzii dokumentu
6. Odkaz na obrazovku editovania dokumentu

6.2 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 pristupujú 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.

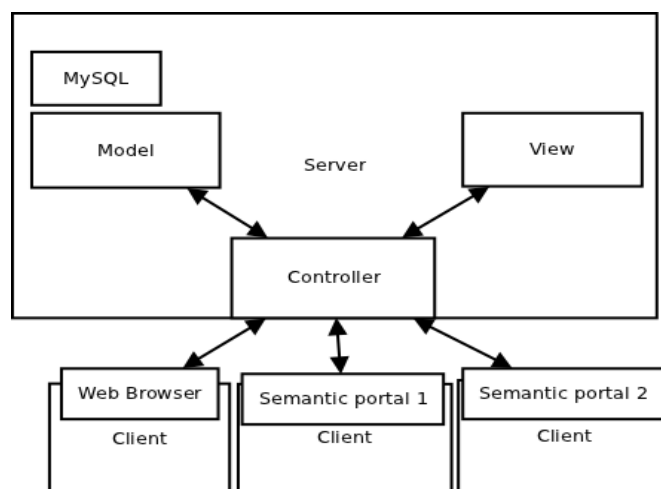
Klientská časť aplikácie je vďaka klientskym skriptom interaktívna a odľahčuje server od niektorých činností (vyhľadávanie a zoradovanie dát). Jej súčasťou je tiež pokročilé zobrazenie dát vo forme grafu ako aj práca s grafom. Klientske skripty sú tiež nápomocné pri zlepšovaní používateľského pohodlia 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. 9. Architektúra systému.

6.3 Dátový model systému

Dátový model sa nachádza na Obr. 10. Zobrazuje entity existujúce v systéme a ich vzájomné vzťahy. Popis modelu je rozdelený na štyri hlavné časti.

6.3.1 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.

6.3.2 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.

6.3.3 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.

6.3.4 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.

6.3.5 Annotation

K dokumentu je možné priradiť 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.

6.3.6 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.

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

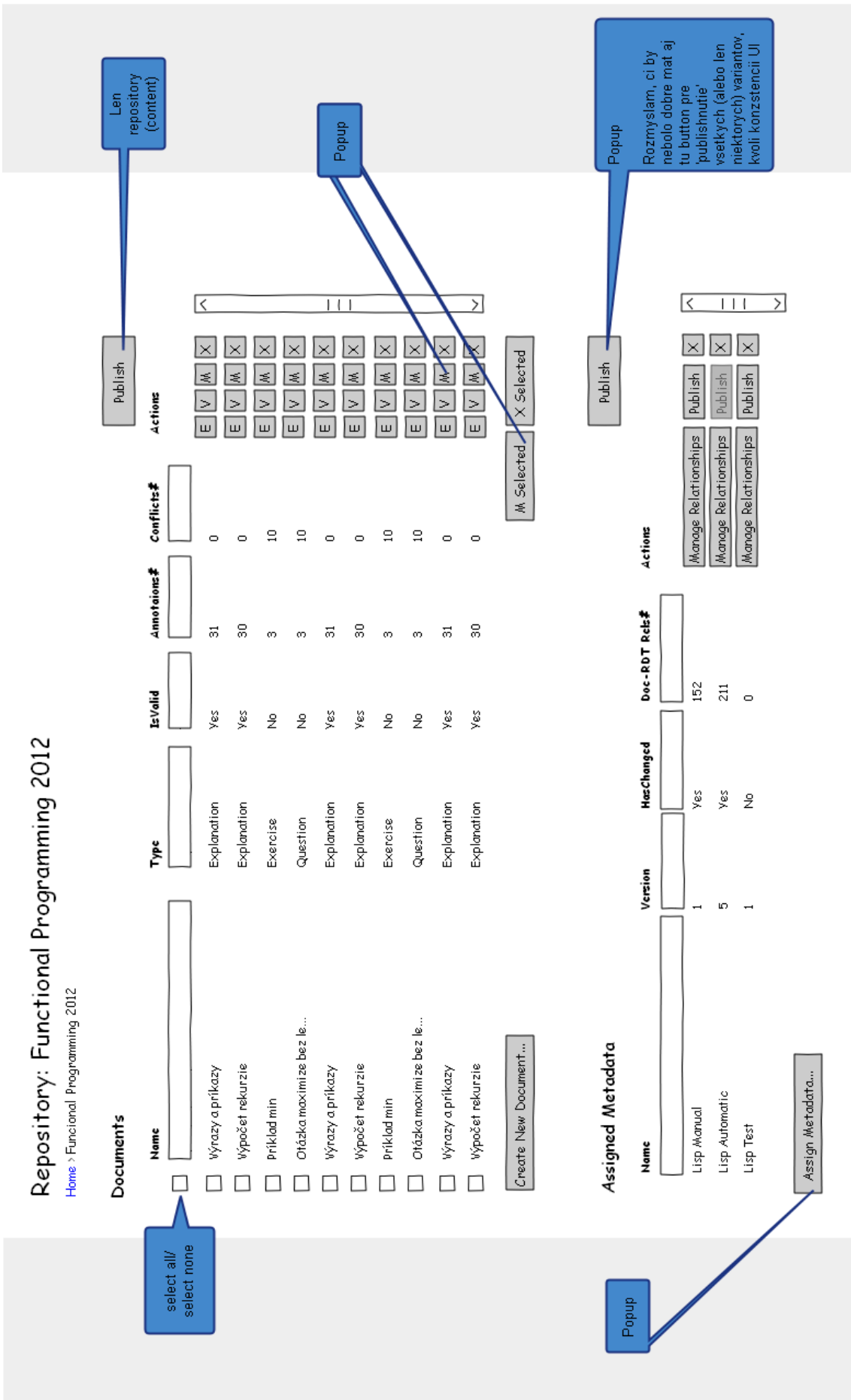
Name	Version	HasChanged	RDT#	RDT Rd#	Repositories#
<input type="checkbox"/> LISp manual	2	yes	301	504	3
<input type="checkbox"/> LISp automatic	1	no	289	578	1
<input type="checkbox"/> LISp new	1	yes	12	0	0
<input type="checkbox"/> PST manual	13	no	531	1003	1
<input type="checkbox"/> PR.OLOG manual	3	yes	281	423	2
<input type="checkbox"/> PR.OLOG 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
<input type="checkbox"/> <input type="checkbox"/> Publish <input type="checkbox"/> X

Popup

Obr. 11. Obrazovka Home.



Obr. 12. 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"/> Question	3	<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	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"/> Question	10	<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"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="select all/ select none"/>
Výrazy a príkazy	MAPCAR	has-rdt	0.33	<input type="button" value="E"/> <input type="button" value="X"/>
Funkcionálny v LISpe	funkcionalí	has-rdt	1.00	<input type="button" value="E"/> <input type="button" value="X"/>
Čistý výraz	výraz	has-rdt	0.90	<input type="button" value="E"/> <input type="button" value="X"/>
Čistý výraz	príkaz	has-rdt	0.70	<input type="button" value="E"/> <input type="button" value="X"/>
<input checked="" type="checkbox"/>	rekurzia	has-rdt	0.05	<input type="button" value="E"/> <input type="button" value="X"/>
Výrazy a príkazy	MAPCAR	has-rdt	0.33	<input type="button" value="E"/> <input type="button" value="X"/>
Funkcional MAPCAR	funkcionalí	has-rdt	1.00	<input type="button" value="E"/> <input type="button" value="X"/>
Funkcional MAPCAR	MAPCAR	has-rdt	0.95	<input type="button" value="E"/> <input type="button" value="X"/>
Vetvenie	COND	has-rdt	0.81	<input type="button" value="E"/> <input type="button" value="X"/>
Vetvenie	forma	has-rdt	0.57	<input type="button" value="E"/> <input type="button" value="X"/>
<input checked="" type="checkbox"/>	MAPCAR	has-rdt	0.33	<input type="button" value="E"/> <input type="button" value="X"/>
Funkcionálny v LISpe	funkcionalí	has-rdt	1.00	<input type="button" value="E"/> <input type="button" value="X"/>
Čistý výraz	výraz	has-rdt	0.90	<input type="button" value="E"/> <input type="button" value="X"/>
Čistý výraz	príkaz	has-rdt	0.70	<input type="button" value="E"/> <input type="button" value="X"/>
<input type="checkbox"/>	rekurzia	has-rdt	0.05	<input type="button" value="E"/> <input type="button" value="X"/>
<input type="checkbox"/>	MAPCAR	has-rdt	0.33	<input type="button" value="E"/> <input type="button" value="X"/>
<input type="checkbox"/>	funkcionalí	has-rdt	1.00	<input type="button" value="E"/> <input type="button" value="X"/>
<input type="checkbox"/>	MAPCAR	has-rdt	0.95	<input type="button" value="E"/> <input type="button" value="X"/>
<input type="checkbox"/>	COND	has-rdt	0.81	<input type="button" value="E"/> <input type="button" value="X"/>
<input type="checkbox"/>	forma	has-rdt	0.57	<input type="button" value="E"/> <input type="button" value="X"/>

autocomplete combobox

Document name
 RDT name
 Type
 Weight

X Selected

inline

Obr. 15. Obrázovka Repo Detail (RDV1).

Document: Výrazy a príkazy

Home > Functional Programming 2012 > Výrazy a príkazy

Edit

View

Document-ID: flip-book-p-2.2

Wrap Text

Full-screen

```
<?xml version="1.0" encoding="UTF-8"?>
<article xmlns="http://docbook.org/ns/docbook"
  role="explanation" xml:id="flip-book-p-2.2"
  version="0" difficulty="0">
  <title>Abstraktný typ údajov lisp-soznam</title>
  <para>Praktické používanie typov údajov si
  vyžaduje ich presný opis – špecifikáciu. Mestačí iba
  vymenovať prípadné operácie, povolené spôsoby ich
  kombinácií do formál (signatúra). Treba opísať aj ich
  sémantiku. Na opis sémantiky sa často používa
  <emphasis>axiomatizácia špecifikácií</emphasis>. V
  axiomatizkej špecifikácii sa sémantika operácií
  opisuje logickými 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. Novoríme preto o <emphasis>abstraktných
  typoch údajov</emphasis> (angl. abstract data type -
  ADT).</para>
  <para>Abstraktný typ údajov sahá na hodnoty a
  elementárne operácie definované nad týmito hodnotami.
  Najskôr budeme uvažovať hodnoty typu lisp-soznam, a
  potom aj operácie nad týmito hodnotami.</para>
  <para>Vzhľadom na to, že soznam v lisp je
  špeciálny prípad abstraktného typu údajov soznam,
  označili sme ho pojmom lisp-soznam. V ďalšom texte
  budeme pod pojmom soznam myslieť vždy lisp-soznam. V
  prípade potreby túto skutočnosť explicitne
  zdôrazníme. Zároveň budeme na
```

Properties?

Annotations

RD Ts

Document Relationships

Files

SelectedText/Context

Content

Author

MacCtrl Actions

Content	Author	Type	MacCtrl Actions
Model umozňuje Model umozňuje	Milan Stručik	Highlight	No
Model umozňuje Model umozňuje	Agáta Impl...	Error report	Yes
SYSGST Identifikatory, napr.: systst a systst V zohem prípadě	Agáta Impl...	Error report	Yes
Softvérové procesy Softvérové procesy	Peter Čičko	Tag	No
2 Softvérové procesy V tejto časti sa venujeme opisu softvérových	Peter Čičko	Tag	No
Model umozňuje Model umozňuje	Milan Stručik	Highlight	No
SYSGST Identifikatory, napr.: systst a systst V zohem prípadě	Agáta Impl...	Error report	Yes
Softvérové procesy Softvérové procesy	Peter Čičko	Tag	No
2 Softvérové procesy V tejto časti sa venujeme opisu softvérových	Peter Čičko	Tag	No

Load XML...

Preview

History

Validate

Save

Full-screen tu ci tu?

Properties, ako Name, LO-ID, atd. Zbytočne zaberaju miesto nad obsahom suboru, možno by som ich dal ako samostatnu tabku 'Properties'

Pop-up

Obr. 16. Obrazovka Document Detail – edit (DE1).

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 *axiomatizácia špecifikácia*. V axiomatizácii 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 axiomatizácii 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. Najskôr budeme uvádzať hodnoty typu lisp-zoznam, a potom aj operácie nad týmito hodnotami. Vzájomom 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 viewer 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 identifikatory, napr.: systst a systst. V záznam prípadě	Error report	Agóda Impl...	Yes
Softvérové procesy	Softvérové procesy. V tejto časti sa venujeme opisu softvérových	Tag	Peter Čečko	No
Model umožňuje	Model umožňuje	Highlight	Milan Stručík	No
sysgst	V tomto texte je preklep identifikatory, napr.: systst a systst. V záznam prípadě	Error report	Agóda Impl...	Yes
Softvérové procesy	Softvérové procesy. V tejto časti sa venujeme opisu softvérových	Tag	Peter Čečko	No

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

Popup

Obr. 17. 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.