

MASARYKOVA UNIVERZITA V BRNĚ
FAKULTA INFORMATIKY



Aplikace pro tvorbu otázek a testů

DIPLOMOVÁ PRÁCE

Tomáš Udržal

Brno, jaro 2005

Prohlášení

Prohlašuji, že tato diplomová práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

Vedoucí práce: RNDr. Tomáš Pitner, Ph.D.

Poděkování

Děkuji RNDr. Tomáši Pitnerovi, Ph.D., vedoucímu mé diplomové práce, za všechny cenný čas a ochotu, se kterou zodpovídal mé dotazy. Velký dík patří také kolegovi Martinu Tomolovi za jeho podnětné rady a připomínky.

Shrnutí

Práce seznamuje čtenáře se standardy určenými pro elektronickou podporu vzdělávání. Popisuje tvorbu vizuálního editoru pro vytváření a úpravu otázek a testů dle zvoleného standardu.

Klíčová slova

test, otázka, testování, e-learning, elektronické vzdělávání, IMS, Java, GUI

Obsah

1	Úvod	1
2	Standardy e-learningového vzdělávání	2
2.1	Komunikační rozhraní nebo API	3
2.2	Meta-data	3
2.3	Výměna obsahu	4
2.4	IMS	4
3	IMS QTI	5
3.1	ASI objektový model	5
3.2	Response Processing a Outcomes Processing	6
3.3	Selection and Ordering	7
3.4	Softwarové nástroje s podporou IMS	7
3.4.1	Macromedia Authorware	7
3.4.2	IMS Package Editor	7
3.4.3	Lersus	8
3.4.4	Respondus	8
4	Použité technologie	9
4.1	Java	9
4.2	ANTLR	10
5	Aplikace	11
5.1	Specifikace požadavků	11
5.2	Analýza a návrh	13
5.2.1	Diagram tříd	13
5.2.2	Tvorby testu a repositáře otázek	14
5.2.3	Podporované typy výběru a uspořádání položek	17
5.2.4	Podporované typy otázek	19
5.2.4.1	Otázka typu True/False	20
5.2.4.2	Otázka typu Multiple Choice	20
5.2.4.3	Otázka typu Multiple Response	21
5.2.4.4	Otázka typu Fill-In-Blank	22
5.2.4.5	Otázka typu Short Answer	22
5.2.4.6	Otázka typu Ordering Objects	23
5.2.4.7	Otázka typu Connecting Objects	23
5.2.4.8	Otázka typu Combined	24
5.2.5	Algoritmy hodnocení	25
5.3	Implementace	27
5.3.1	Struktura tříd	27
5.3.2	Rozhraní Loader a Saver	28
5.3.3	Zpracování odpovědí	28
5.3.3.1	Gramatika pro tvorbu výrazů hodnocení odpovědí	29
5.3.4	Možnosti vkládání multimédií	32

5.3.5	Internacionalizace	33
5.3.6	Vytvoření manuálu a jeho propojení s aplikací	34
5.3.7	Java Web Start	34
6	Závěr	35
	Bibliografie	36
	Rejstřík	37
A	Ukázka aplikace	38
B	Význam symbolů gramatiky	40
C	Obsah přiloženého CD	41
D	ANTLR software licence	42
E	Dom4j software licence	43

Kapitola 1

Úvod

V posledních letech vzniká množství LMS systémů (Learning Management System) neboli systémů pro řízení výuky, což je typicky softwarová aplikace určená pro řízení výukových aktivit v organizaci. Spolu s nimi vznikají specifikace pro definování jednotlivých částí výuky a vztahů mezi nimi. Tyto specifikace popisují např. způsob organizace lekcí, ukládání jejich zdrojů, vytváření testů, hodnocení studentů či záznam výsledků.

Tato práce se zaměřuje na oblast vytváření testů a samozřejmě také otázek. Nejprve budou popsány existující standardy e-learningového vzdělávání a jejich klíčové vlastnosti. Dále následuje popis vybraného standardu IMS a krátce je uveden přehled vlastností současných softwarových produktů, které tento standard podporují.

Hlavní část práce popisuje způsob vytvoření autorské desktopové aplikace v programovacím jazyce Java, která díky grafickému uživatelskému rozhraní umožňuje snadné vytváření nebo editaci testových otázek. Při popisu jednotlivých částí aplikace bude kladen důraz na dodržení postupu vývoje softwarového nástroje a bude poukázáno na to, které prvky standardu IMS byly do jaké míry implementovány.

Aplikace používá při ukládání objektů do souboru nebo při jejich načítání či importu aplikační rozhraní standardu IMS (dále jen API IMS) vytvářeným souběžně s touto aplikací kolegou Martinem Tomolou [Tomola05].

Kapitola 2

Standardy e-learningového vzdělávání

Před tím, než se e-learningové metody stanou „standardem“, nazývají se „specifikace“. Specifikace jsou posuzovány některou z akreditovaných organizací, jakými jsou třeba ISO, ANSI nebo IEEE. Problém těchto organizací spočívá v tom, že nejsou schopni rychle reagovat na nové potřeby, neboť proces standardizace běžně trvá několik let. Proto existují kromě nich také specifikační organizace, které vydávají různá doporučení a návody v nepoměrně kratší době. Následující souhrny popisují nejznámější specifikační organizace a jejich klíčový přínos pro e-learning.

Advanced Distributed Learning (ADL) Tato organizace zřízená americkým ministerstvem obrany má za úkol zajistit přístup ke kvalitním zdrojům vzdělávání, které budou snadno dostupné a mohou být upraveny individuálním potřebám. ADL¹ působí v oblasti standardizace jako most mezi průmyslovými a akademickými konsorciemi a standardizačními organizacemi s obecným zaměřením. Nejrozšířenější podobou ADL je specifikace SCORM (Sharable Content Object Reference Model) založená na existujících standardech IMS, IEEE, AICC a udržovaná tak, aby byla trvale s těmito standardy kompatibilní. SCORM poskytuje konkrétní příklady, praktická použití a vysvětlení, která pomáhají vývojářům implementovat specifikace elektronického vzdělávání.

Aviation Industry CBT Committee (AICC) AICC² je jeden z prvních standardů, který specifikuje např. jak má fungovat výměna výukových materiálů mezi kurzy a systémy nebo způsob uchování dat o výsledcích zkoušení atd. LMS (Learning Management System) může v souladu s tímto standardem podstoupit certifikaci nebo může pouze proklamovat, že byl vytvořen podle požadavků specifikace, aniž by byl posouzen.

IEEE Learning Technology Standards Committee (IEEE LTSC) IEEE³ je mezinárodní organizace, která vyvíjí standardy a doporučení pro elektrické, elektronické, počítačové a komunikační systémy. Nejrozšířenější IEEE LTSC specifikací je Learning Object Meta-data (LOM), která popisuje skupiny prvků a prvky, které popisují zdroje

1. <http://www.adlnet.org/>

2. <http://www.aicc.org/>

3. <http://ltsc.ieee.org/>

2.1. KOMUNIKAČNÍ ROZHRANÍ NEBO API

elektronického vzdělávání. Nevýhodou standardu je placený přístup k jeho plnému znění.

IMS Global Consortium (IMS) IMS⁴ je konsorcium, které se zaměřuje na vývoj specifikací založených na XML. Specifikace popisují klíčové vlastnosti kursů, hodin, testů, učitelů a skupin, navíc poskytují strukturu pro reprezentaci meta-dat (tedy dat o datech). Popis e-learningových zdrojů umožňuje jejich vyhledávání, výměnu a správu. Nejznámějšími IMS specifikacemi jsou IMS Meta-data, IMS Content Packaging a IMS QTI (Question and Test Interoperability).

E-learningové organizace se zaměřují na tři hlavní oblasti:

- komunikační rozhraní nebo API - jak zdroje komunikují s ostatními systémy
- meta-data - jak popsat e-learningové zdroje
- výměna obsahu - přenositelnost mezi různými systémy

2.1 Komunikační rozhraní nebo API

Komunikační rozhraní je způsob, jakým si zdroje dynamicky vyměňují informace. Aktuální stav je takový, že specifikace AICC a ADL specifikují komunikační rozhraní, zatímco IEEE a IMS ještě toto rozhraní explicitně neidentifikovaly.

Nejrozšířenějším komunikačním protokolem pro vývojáře obsahu je Hypertext AICC Communication Protocol (AICC HACP). Obsahuje záznamy o vykonané práci studenta a historii lekcí, což je užitečné při zobrazení práce studenta v daných obdobích. ADL specifikuje alternativu k AICC HACP nazvanou „API Communication“. Hlavním rozdílem mezi těmito dvěma pohledy je velikost množiny datových elementů, se kterými lze komunikovat; ADL používá podmnožinu elementů používanými AICC.

2.2 Meta-data

Meta-data poskytují možnost popisu jakýchkoli e-learningových zdrojů. Lze např. uchovávat informace o konzultačních hodinách, krátký popis obsahu, jazyku atd. Tyto informace lze používat pro vyhledávání nebo návrh, vývoj a správu životního cyklu.

Nejrozšířenějším standardem v této oblasti je IEEE Learning Object Meta-data (LOM). Standard LOM používá pro definici výukového objektu velice široký pojem, který umožňuje široké použití. Jeho vadou je však nepovinnost elementů, kterými definuje objekt, což není příliš vhodné pro praktické použití. LOM je tedy příkladem standardu, který se v původní podobě téměř nepoužívá, ale jehož základy jsou obsaženy v jiných specifikacích, např. IMS a ADL.

4. <http://www.imsglobal.org>

2.3 Výměna obsahu

Výměna obsahu znamená na nižší úrovni přenositelnost testu k uživateli a zpět. Nejrozšířenější specifikací pro první zmíněný směr je IMS QTI a pro opačný směr, zpracování výsledků, IMS Results Reporting. IMS QTI je poměrně složitá specifikace a většina nástrojů, které podporují IMS QTI, umí pracovat pouze s podmnožinou všech možných typů otázek, což však nevadí, aby dnes IMS QTI nebyla využívána čím dál častěji.

Na vyšší úrovni znamená výměna obsahu přenositelnost mezi různými systémy, v této oblasti hraje klíčovou roli pojem „Content-Packaging“. Content Packaging znamená shromáždění a popis zdrojů pro daný postup. Například specifikace IMS QTI může sdružovat jednotlivé položky do svazků a nakonec do testu. V širším pojetí může být proces balení použit pro sdružení několika QTI souborů do katalogu testů nebo do kurzu, který obsahuje více testů (IMS Content Packaging).

Nejnámějšími specifikacemi jsou AICC Course Structure File (CSF) a IMS Content Packaging Specification. CSF implementace používá pro popis Windows INI a CSV soubory, zatímco IMS používá XML formát.

2.4 IMS

Jak je zřejmé z výše uvedeného přehledu standardů elektronického vzdělávání, nejobsáhlejším standardem je IMS. Uvedeme několik důvodů, proč je v současné době IMS nejvíce používaným standardem:

- specifikuje klíčové vlastnosti potřebné v oblasti elektronické podpory výuky
- ve většině oblastí hraje dominantní roli (Meta-data, Content Packaging, Question & Test Interoperability, Results Reporting)
- používá značkovací jazyk XML
- na webových stránkách je obsáhlá dokumentace specifikace včetně DTD, XML schémat a praktických příkladů

Kapitola 3

IMS QTI

IMS Question & Test Interoperability specifikace popisuje základní strukturu pro reprezentaci otázek a testů, která je realizována ve formátu XML, čímž je zaručena jejich plná přenositelnost. V této kapitole uvedeme klíčové pojmy, které IMS QTI používá, blíže je rozebereme a uvedeme některé softwarové nástroje deklarující podporu IMS QTI, které jsou v současnosti na trhu.

3.1 ASI objektový model

Nejdůležitějším objektem je Assessment, který je obdobou testu. Jeho základními kameny jsou položky (Item). Navíc, Assessment může být složen z bloků položek, které jsou logicky podobné. Tyto skupiny jsou definované jako sekce (Section) a Assessment se tak může skládat z jedné nebo více sekcí, jenž samotné mohou být složeny z položek nebo dalších sekcí. Dohromady, tyto tři datové objekty jsou nazývány jako ASI struktury (Assessment, Section, Item). Tyto objekty mohou být svázány dohromady a tvořit Object Bank neboli repozitář otázek. Tento Object Bank může být externě referencován a používán jako jeden objekt. Aby předešla omezením spojeným s užitím slov jako uživatel, student nebo učitel, IMS QTI specifikace užívá termínu účastník (Participant) k označení osoby pracující s Assessment. Čili, klíčové definice jsou:

Položka (Item) nejmenší vyměnitelný objekt v QTI. Položka je více než otázka, neboť obsahuje otázku, instrukce k prezentaci/zobrazení, zpracování odpovědi na základě uživatelově odpovědi, reakce, která může být přítomna (včetně nápovědy a řešení) a meta-data popisující položku

Sekce (Section) je použita k vytváření hierarchických hodnotících objektů. Sekce může obsahovat jednu nebo více položek a/nebo sekcí. Sekce je užívána ze dvou různých důvodů, buď pro zobrazení rozdílných skupin užitých odpovídajícím vzdělávacím paradigmatickým (např. může odpovídat tématu předmětu biologie) nebo k omezení míry opakování instrukcí a ke kontrole způsobů, kterými mohou být různé posloupnosti vytvářeny (možnost nastavení výběru a pořadí položek)

3.2. RESPONSE PROCESSING A OUTCOMES PROCESSING

Test (Assessment) obsahuje soubor položek, které jsou určeny k rozlišení stupně obtížnosti, nebo jinak, kteří uživatelé mají přístup ke kterým objektům. Assessment obsahuje všechny potřebné instrukce k umožnění rozdílných posloupností položek a korespondujících shromážděných skóre k vytvoření celkového skóre. Může se skládat z jedné nebo více sekcí

Repositář otázek (Object Bank) je kolekce položek, sekcí nebo jejich složeninou. Repositář otázek má svůj jedinečný identifikátor a meta-data umožňující prohledávání jeho obsahu. Obvykle obsahuje databázi objektů, které mohou použity k vytváření testů

Položka je základním kamenem specifikace, kromě dalších informací, které budou popsány v následující podkapitole, obsahuje také samotnou otázku a odpovědi. IMS QTI podporuje celkem 12 základních druhů otázek, kromě nich však umožňuje vytvářet i libovolně složité otázky vzniklé kombinací ze základních druhů.

Obsahem kromě textu mohou být také další prvky jako obrázky, videa, zvukové nahrávky, applety či dokonce interaktivní aplikace. QTI umožňuje i použití tzv. alternativního obsahu, čímž lze dosáhnout např. vícejazyčného obsahu. Přestože QTI umožňuje určité formátování textu či určování velikosti a pozice prvků, konkrétní *zobrazení* otázek či testu *závisí na zobrazovacím systému*, který bude interpretovat informace obsažené v XML souboru. QTI styl *zobrazení* *nedefinuje*, pouze poskytuje doporučení zobrazovacímu systému.

3.2 Response Processing a Outcomes Processing

QTI obsahuje propracovaný systém pro vyhodnocování otázek. U každé otázky je možné nadefinovat si proměnné, do kterých se zaznamenává ohodnocení za odpověď na otázku (Response Processing). Pokud si žádné vlastní proměnné nenadefinujeme, pracuje se se standardně generovanou proměnnou. Dalším krokem je vytvoření logických podmínek, při jejichž splnění se změní hodnota námi určené proměnné. Tyto podmínky mohou být tvořeny logickými či relačními operátory a umožňují tak vytvoření libovolného výrazu v podmínce. U každé podmínky QTI umožňuje nastavení zpětné vazby k otázce definováním reakce (Feedback), nápovědy (Hint) či řešení (Solution).

V širším kontextu QTI používá hodnotící algoritmy (Outcomes Processing), které vyhodnocují body z jednotlivých položek, a pro jejichž výsledné hodnoty lze vytvořit pro testování hodnoty výsledku sadu podmínek. Pro každou z těchto podmínek pak lze definovat odpovídající reakci (Feedback) podobně jako pro podmínky u položek. QTI obsahuje sadu předdefinovaných vyhodnocovacích algoritmů, které pracují se standardně generovanými proměnnými, a umožňuje také tvorbu vlastních algoritmů, které budou pracovat s vlastními definovanými proměnnými.

3.3 Selection and Ordering

QTI obsahuje mechanismus výběru a uspořádání (Selection and Ordering), který slouží k vymezení pravidel pro konečné zobrazení testu zkoušenému. Skládá se ze tří částí: výběr, uspořádání a sekvence. Výběr spočívá v aplikaci výběrových pravidel na položky, přičemž je opět možno používat logické asociace. Uspořádání definuje pořadí vybraných otázek účastníkovi (např. náhodné) a sekvence určuje počet zobrazení jednotlivé položky.

3.4 Softwarové nástroje s podporou IMS

V této části uvedeme softwarové nástroje s podporou specifikace IMS, které jsou v současné době dostupné.

3.4.1 Macromedia Authorware

Vlastnosti¹:

- interaktivní nástroj pro tvorbu interaktivního multimediálního obsahu
- podporuje audio streaming, import Flash a Powerpoint, XML import a export
- lze exportovat do SCORM modulu
- komerční produkt

Authorware se příliš nepoužívá pro vytváření e-learningových kurzů, hlavními důvody jsou vysoká cena, proprietární datový formát a špatná podpora pro vytváření webového obsahu.

3.4.2 IMS Package Editor

Vlastnosti²:

- import/tvorba balíku lekce založená na specifikaci IMS Content Packaging
- organizace lekcí různými způsoby
- detekce a kopírování přidružených zdrojů (odkazované či vložené soubory) v lekcích do balíčku
- GUI editování pro meta-data založené na specifikaci IMS Meta-data
- freeware

1. <http://www.macromedia.com/software/authorware/>

2. <http://www.met.ed.ac.uk/pac-man/editor/>

3.4.3 Lersus

Vlastnosti³:

- tvorba interaktivních tutoriálů, CBT kurzů a WBT kurzů dle existujících e-learningových standardů
- obsah je vytvářen ve formě webové stránky nebo v tištěné podobě
- komerční produkt

3.4.4 Respondus

Vlastnosti⁴:

- tvorba kvízů, testů a průzkumů
- import otázek z vnějšího textu (ne všechny druhy) nebo z WebCT
- správa většího počtu otázek do „question bank“
- umožňuje vkládat obrázky, audio, video
- obsahuje matematický editor
- generuje zprávy o výsledcích
- komerční produkt

3. <http://www.lersus.com>

4. <http://www.respondus.com>

Kapitola 4

Použité technologie

4.1 Java

Java je objektivě orientovaný programovací jazyk vyvinutý společností Sun Microsystems, Inc.¹ Jeho klíčové vlastnosti jsou:

jednoduchý vychází z C++, vynechává prvky, které jsou málo používané, zavádějící nebo těžko pochopitelné (přetypování operátorů, vícenásobná dědičnost) a přidává automatický garbage-collection

distribuovaný obsahuje knihovny pro přístup k TCP/IP protokolům pro přístup k objektům na síti

nezávislý na architektuře, interpretovaný a přenositelný umožňuje vytvářet aplikace pro více platforem. Překladač generuje interpretovaný bytekód, který je nezávislý na procesoru a lze jej vykonávat kdekoli, kde běží Java Virtual Machine. Navíc specifikace Javy neobsahuje žádné implementačně závislé prvky, např. velikosti, kódování a aritmetika základních datových typů odpovídá standardům, které používají dnešní procesory

robustní programy jsou spolehlivé. Java věnuje zvýšenou pozornost odhalování možných problémů programových konstrukcí, dynamické kontrole za běhu. Implementuje mechanismu výjimek pro zpracování chyb za běhu programu

bezpečný obsahuje bezpečnostní mechanismy, jako např. podepisování kódu veřejným klíčem nebo přidělování práv pro různé akce

1. <http://www.sun.com>

multithreadový obsahuje technologii umožňující paralelní běh více větví programu současně. Součástí jazyka jsou monitory - synchronizační nástroje

Java obsahuje standardní knihovny, s jejichž množstvím se nemůže srovnávat žádný jiný programovací jazyk. K dispozici jsou knihovny pro tvorbu grafického uživatelského rozhraní (GUI), vstupně/výstupními proudy, práci s textem, komunikaci s databázemi, práci s komprimovanými soubory atd.

4.2 ANTLR

ANTLR² (ANOther Tool for Language Recognition) je jedním z nástrojů, který poskytuje rámec pro konstrukci analyzátorů, překladačů a převaděčů z gramatických popisovačů obsahujících Java, C#, Python nebo C++ akce. ANTLR obsahuje konzistentní syntax pro specifikaci slovníků, parserů a stromových parserů a umožňuje tak překlad, konstrukci stromu a jeho procházení. Pomocí nástroje ANTLR byly vytvořeny parsery takových jazyků jako Java, GNU C, HTML či Oracle 7 SQL.

Aplikace používá ANTLR k vytvoření parseru, který ověřuje uživatelem zadané podmínky pro hodnocení různých odpovědí, zdali jsou správně strukturované. ANTLR pracuje se souborem s definicí gramatiky, který může obsahovat gramatiku a slovník včetně instrukcí pro zpracování; ANTLR pak zajistí vytvoření tříd v jazyce Java, které jsou přímo použitelné.

2. <http://www.antlr.org>

Kapitola 5

Aplikace

Cílem této práce bylo vytvoření autorské desktopové aplikace pro usnadnění práce učitelům při vytváření nebo editaci testových otázek. Za tímto účelem byla primárně vytvořena struktura vnitřních javovských tříd pro ukládání objektů za běhu programu (tj. testu, otázek, odpovědí, reakcí atd.), přičemž tato struktura odpovídala logickému uspořádání objektů standardu IMS. S výhodou zde bylo použito charakteristických prvků jazyka Java a to dědičnosti a zapouzdření.

Dalším krokem vývoje bylo vytvoření tříd grafického uživatelského rozhraní, které pohodlně umožňuje uživateli vytvářet nebo editovat testy či repositáře otázek, přidávat nebo upravovat otázky. Při tvorbě samotného grafického vzhledu bylo výrazným ulehčením použití nástroje Netbeans 4.0 a jeho GUI Builderu.

Další fází bylo vytvoření manuálu k aplikaci včetně jejich propojení a nakonec byla aplikace internacionalizována do českého a anglického jazyka. V následujících kapitolách nahlédneme na uvedené kroky z teoretického hlediska a blíže ukážeme, jakým způsobem aplikace vznikala.

Proces tvorby aplikace odpovídal vodopádovému modelu softwarového procesu, tedy vývoj se skládal z následujících kroků:

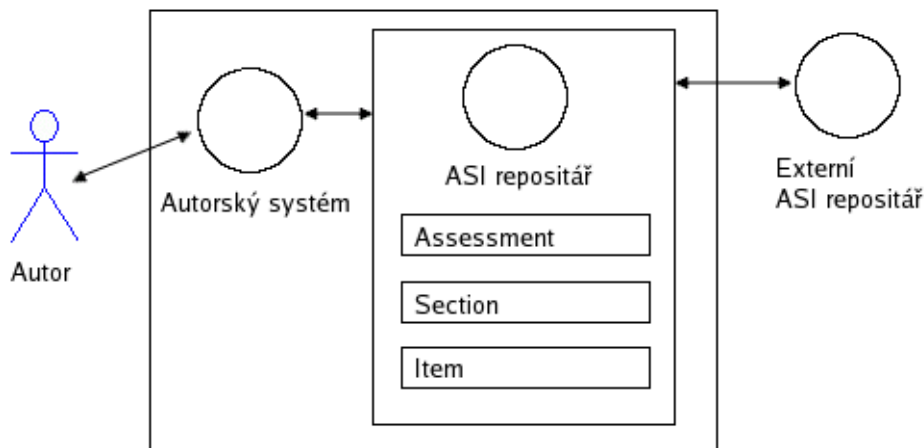
- Specifikace požadavků
- Analýza a návrh softwarového systému
- Implementace (kódování)

Výše uvedené kroky budou rozčleněny do kapitol, které odpovídají těmto třem fázím vývoje. Každá z těchto fází používá specifické modely a diagramy pro popis chování systému. Jednotlivé kapitoly začnou popisem v teoretické rovině následovaným konkrétní ukázkou funkčnosti aplikace, jakým způsobem zajišťuje požadované chování.

5.1 Specifikace požadavků

Cílem specifikace požadavků je popsat, co má softwarový systém dělat prostřednictvím specifikace jeho funkcionality [Vondrak02]. Modely, které jsou v rámci specifikace činností vytvářeny, vychází z tzv. případů užití (Use Cases) tvořených:

- aktéry definující uživatele či jiné systémy, kteří budou vstupovat do interakce s vyvíjeným softwarovým systémem
- případy užití specifikující vzory chování realizovaných softwarovým systémem. Každý případ užití lze chápat jako posloupnost transakcí vykonaných v dialogu mezi aktérem a softwarovým systémem



Obrázek 5.1: Diagram případů užití

V našem případě (obrázek 5.1) se vyskytují v diagramu komponenty:

- Autor - tvůrce, který spouští autorský systém
- Autorský systém - proces, který podporuje vytváření nebo editování testů, sekcí nebo položek (ASI struktur)
- ASI repositář - úložiště lokálních ASI struktur
- Externí ASI repositář - úložiště externích ASI struktur, které mohou být importovány za použití QTI specifikace

Náš diagram tedy umožňuje pouze autorský případ užití, jehož sekvence procesů je následující:

- Autor spustí autorský systém
- Poté Autor vytváří nebo modifikuje položky, sekce a testy, tyto potom odpovídají QTI specifikaci a jsou uloženy do externího úložiště.

- Autor může také importovat existující ASI struktury, které budou použity pro tvorbu nových. Tyto importované také budou odpovídat QTI specifikaci
- Autor tedy odpovídá za způsob použití vzniklých ASI struktur. Podle vzdělávacího účelu nastavuje potřebné vlastnosti jako seskupování do sekcí, zvolení hodnotícího algoritmu, druh výběru a uspořádání, vlastnosti testu, váhu položek apod.

5.2 Analýza a návrh

Cílem analýzy a návrhu softwarového produktu je ukázat, jakým způsobem bude produkt realizován v implementační fázi [Vondrak02]. Zde již přicházejí na řadu diagramy tříd a diagramy aktivit:

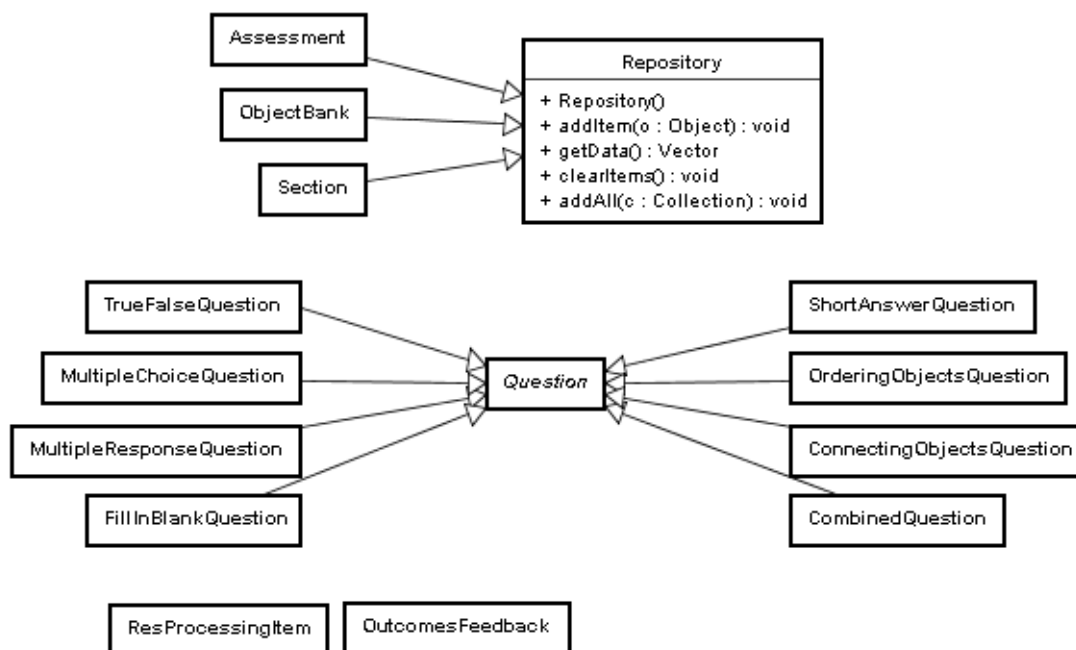
- Diagramy tříd definují základ statického popisu softwarového systému, neboť dokáží postihnout všechny dostupné konfigurace objektů
- Diagramy aktivit popisují proces pomocí jeho stavů reprezentovaných vykonáváním aktivit a pomocí přechodů mezi těmito stavy způsobených ukončením těchto aktivit. Účelem diagramu aktivit je blíže popsat tok činností daný vnitřním mechanismem jejich provádění

5.2.1 Diagram tříd

Diagram tříd na obrázku 5.2 kopíruje ASI strukturu objektů a umožňuje tak snadné procházení jejich strukturou při ukládání nebo načítání do/z XML souboru.

Základní třídou je `Repository`, která obsahuje pouze metody pro práci se seznamem položek, který je v ní obsažen. Jejími potomky jsou třídy `Assessment`, `ObjectBank` a `Section`, které umožňují hierarchii objektů. `Assessment` tak může obsahovat několik sekcí, které mohou obsahovat již konkrétní otázky nebo další sekce. `Assessment` obsahuje následující atributy:

- název testu
- dobu trvání testu
- zvolený algoritmus hodnocení
- zvolený druh výběru a uspořádání položek
- informace o tom, zdali jsou povoleny u položek reakce, nápovědy a řešení
- informace o tom, zdali je povoleno u položek nastavování jejich časové prodlevy
- informace o tom, zdali jsou povoleny u testu podmínky pro hodnocení algoritmu



Obrázek 5.2: Celkový diagram tříd

- podmínky pro ohodnocení testu (seznam objektů `OutcomesFeedback`)

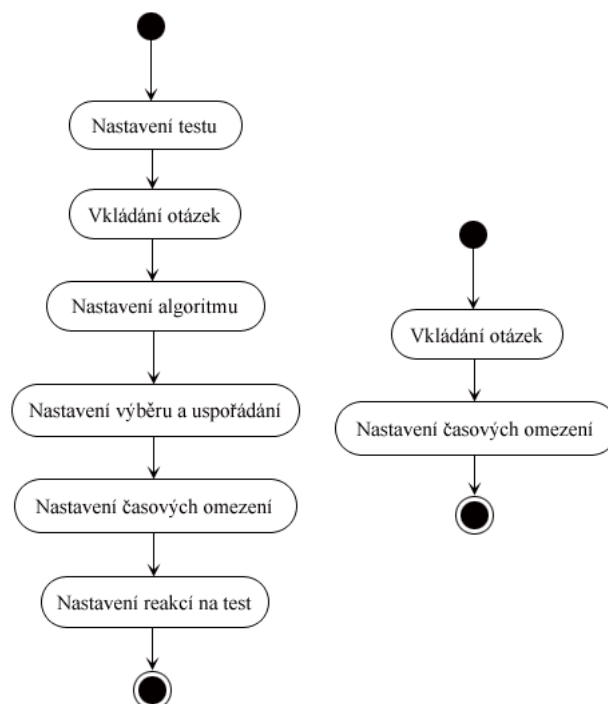
`ObjectBank` je naopak vlastně jenom jiným pojmenováním třídy `Repository`, neboť žádné další metody neobsahuje; `ObjectBank` slouží pouze pro uchování otázek. `Section` obsahuje atributy pro uchování informací o konkrétním druhu výběru a uspořádání, více viz. kapitola 5.2.3.

Jednotlivé otázky jsou přidávány do seznamu objektů v `Repository` (je jedno zdali se jedná o potomka `Assessment`, `ObjectBank` či `Section`). Všechny 8 druhů otázek je odvozeno od abstraktní třídy `Question`, která obsahuje atributy společné pro všechny otázky (typ, název, znění otázky apod.). Obsahuje také seznam objektů `ResProcessingItem`, které obsahují informace o vytvořených výrazech hodnocení odpovědí, reakcí na ně a bodovým ohodnocením.

5.2.2 Tvorby testu a repositáře otázek

Při tvorbě testu nebo repositáře otázek aplikace používá stylu průvodce, který vede uživatele procesem tvorby a postupně mu umožní nastavovat všechny informace a data potřebné pro vytvoření požadovaného objektu. Průvodce obsahuje tlačítka `Zpět` a `Další`, která umožňují postupné vytváření i zpětnou kontrolu či změnu dosavadních nastavení. Průvodce je poměrně obsáhlý, může v případě tvorby testu obsahovat až 6 kroků, které záleží na na-

stavených výchozích vlastnostech testu (v případě tvorby repositáře otázek pouze 2 kroky). Přehled všech možných kroků poskytuje diagram aktivit 5.3 a tabulka 5.2.2.



Obrázek 5.3: Diagramy aktivit procesů Tvorba testu a Tvorba repositáře otázek

Přibližme si nyní jednotlivé kroky.

Nastavení vlastností testu Krok se objeví v případě vytváření nového testu. Umožňuje kromě vložení názvu testu další možnosti nastavení, která ovlivní další chování aplikace. Je to možnost povolení reakcí na otázku (resp. reakce na každý z hodnotících výrazů), povolení nápovědy či řešení u otázky. Dále povolení časových omezení, a to jak celkového testu, tak otázek případně sekcí, a povolení celkového hodnocení testu. Uživatel také může zvolit jeden z algoritmů hodnocení a musí zvolit jeden z druhů výběru a uspořádání.

Vkládání otázek Krok je dostupný v každém případě nezávisle na tom, který objekt uživatel vytváří. Uživatel má možnost vytvářet 8 druhů otázek či může importovat další otázky z externího souboru. V případě tvorby testu již hrají roli nastavení provedená ve výchozím kroku, tedy zdali jsou povoleny reakce, řešení a nápovědy, u výrazů je umožněno nastavování bodů, pokud to vyžaduje nějaký z algoritmů hodnocení.

Tato fáze je časově nejnáročnější, uživatel musí u každé otázky nastavit v případě potřeby další informace potřebné pro algoritmus hodnocení jako např. reakce, nápovědu, řešení. Pro urychlení práce je přítomno zaškrtačací políčko „automatické vyhodnocení“, které se snaží nastavit tyto pokročilé informace na výchozí hodnoty samo.

Nastavení vlastností algoritmu Krok je dostupný pouze při tvorbě testu a pokud to vyžaduje algoritmus hodnocení testu. Jedná se o algoritmy, které používají vážené otázky, nebo o algoritmus Best K of N.

Nastavení výběru a uspořádání Krok je dostupný pouze při tvorbě testu. Nastavují se zde vlastnosti zvoleného druhu výběru a uspořádání, například počet náhodných položek, které mají být vybrány, nebo rozdělení do sekcí podle tématu a zvolení počtu položek, které mají být z každé sekce vybrány.

Nastavení časových omezení Krok je dostupný při tvorbě repositáře otázek nebo při tvorbě testu, pokud byly časová omezení povoleny. Tento krok umožňuje nastavení dobu trvání dané otázky, testu, případně sekce. Údaje se zadávají v minutách, které potom API IMS převádí na hodnotu dle standardu ISO8601.

Nastavení reakcí na test Krok je dostupný pouze při tvorbě testu a pokud byla tato volba povolena. Umožňuje vytvoření výrazů a reakcí, které se vztahují k dosaženému celkovému skóre testu. Lze tak nastavit hodnocení typu prospěl/neprospěl.

	Test	Repositář otázek
Nastavení vlastností testu	ano	
Vkládání otázek	ano	ano
Nastavení vlastností algoritmu	?	
Nastavení výběru a uspořádání	ano	ano
Nastavení časových omezení	?	
Nastavení reakcí na test	?	

Tabulka 5.1: Možnosti aplikace při tvorbě testu nebo repositáře otázek. ? vyjadřuje, že zobrazení tohoto kroku závisí na předchozím nastavení

5.2.3 Podporované typy výběru a uspořádání položek

Specifikace „Selection & Ordering“ obsahuje popis, jak může být kontrolováno pořadí, ve kterém jsou prezentovány sekce a/nebo položky *za běhu*, tedy např. při pozdějším průběhu testu. Proces výběru a uspořádání je operace, ve které jsou nejprve podle nějakých kritérií (např. meta-dat) vybrány položky a poté je určeno jejich pořadí. IMS specifikace podporuje libovolné zanořování sekcí a položek, Assessment či Object Bank může obsahovat Section nebo Item, Section může obsahovat další Section nebo Item.

Specifikace nabízí přes 20 různých možností jak zajistit výběr a uspořádání položek spolu s praktickými ukázkami. Pro potřebu aplikace bylo vybráno 8 nejběžnějších typů:

Sekvenční Otázky jsou zobrazeny v pořadí, v jakém jsou zadány. Každá otázka je zobrazena pouze jednou

Náhodné Otázky jsou zobrazeny v náhodném pořadí, pořadí je pokaždé jiné. Každá otázka je zobrazena pouze jednou

Náhodných X Množina X otázek je vybrána z celkového počtu otázek. Výběr i pořadí je pokaždé jiné. Každá otázka je zobrazena pouze jednou

Náhodné + fixní pozice Otázky jsou zobrazeny v náhodném pořadí, kromě několika označených jako fixní. Každá otázka je zobrazena pouze jednou. Fixní položky rozdělují množinu všech otázek na intervaly a v nich jsou otázky zobrazeny v náhodném pořadí

Náhodných X + fixní pozice Otázky jsou zobrazeny v náhodném pořadí, kromě několika označených jako fixní. Každá otázka je zobrazena pouze jednou. Fixní položky rozdělují množinu všech otázek na intervaly a pro každý z těchto intervalů se volí počet X náhodně vybraných

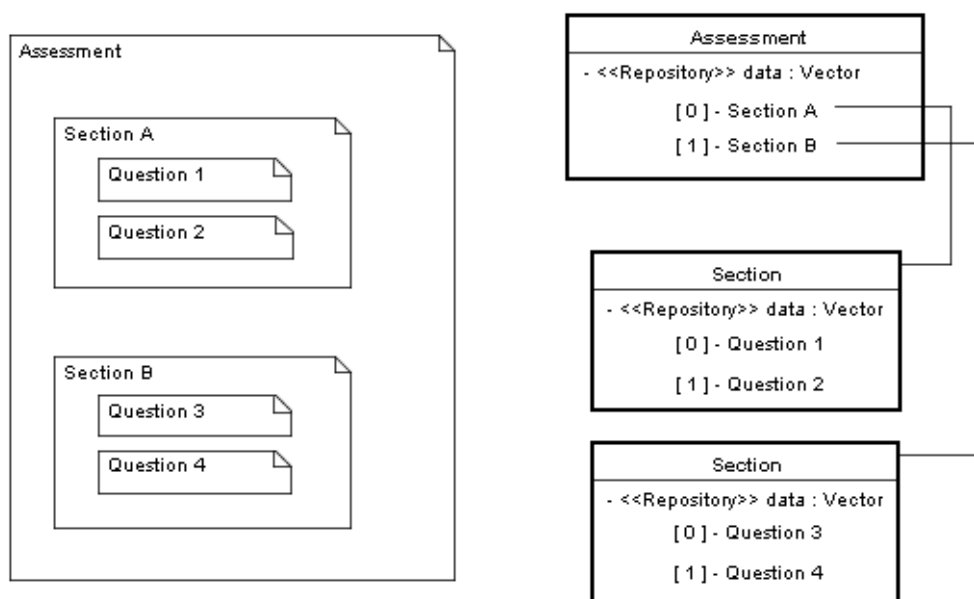
Náhodné podlé tématu Vyberou se otázky se zvoleným tématem, tyto budou zobrazeny v náhodném pořadí. Každá otázka je zobrazena pouze jednou

Náhodných X podle tématu Vyberou se otázky se zvoleným tématem, z těchto bude X náhodně vybráno a zobrazeno v náhodném pořadí. Každá otázka je zobrazena pouze jednou

Seskupení do sekcí podle tématu Otázky jsou seskupeny do sekcí podle tématu. Pro každou z nich je zvoleno X náhodně vybraných a zobrazených. Ve výsledku bude zachováno zvolené pořadí sekcí. Každá otázka je zobrazena pouze jednou

Aplikace zajišťuje volbu jednoho z těchto typů a nastavení jeho vlastností. Výběr a uspořádání umožňuje nastavit pouze u testu, repositář otázek výběr a uspořádání otázek nepodporuje.

Pro uložení struktury sekcí a otázek jsou použity třídy `Repository`, `Assessment`, `ObjectBank` a `Section` (viz. obrázek 5.2). Třídy `Assessment`, `ObjectBank` a `Section` jsou potomky třídy `Repository`, která obsahuje pouze seznam položek, který může obsahovat otázku nebo sekci. Tím dosáhneme libovolné úrovně hierarchizace, pro naše účely však stačí úroveň 2, tedy případ kdy jsou v sekci zanořeny podsekcce. Na obrázku 5.4 je příklad uspořádání do sekcí a jeho reálné uložení do objektů aplikace.



Obrázek 5.4: Uspořádání do sekcí a odpovídající struktura objektů

Třída `Section` obsahuje atributy potřebné pro uchování informací o zvoleném druhu výběru a uspořádání:

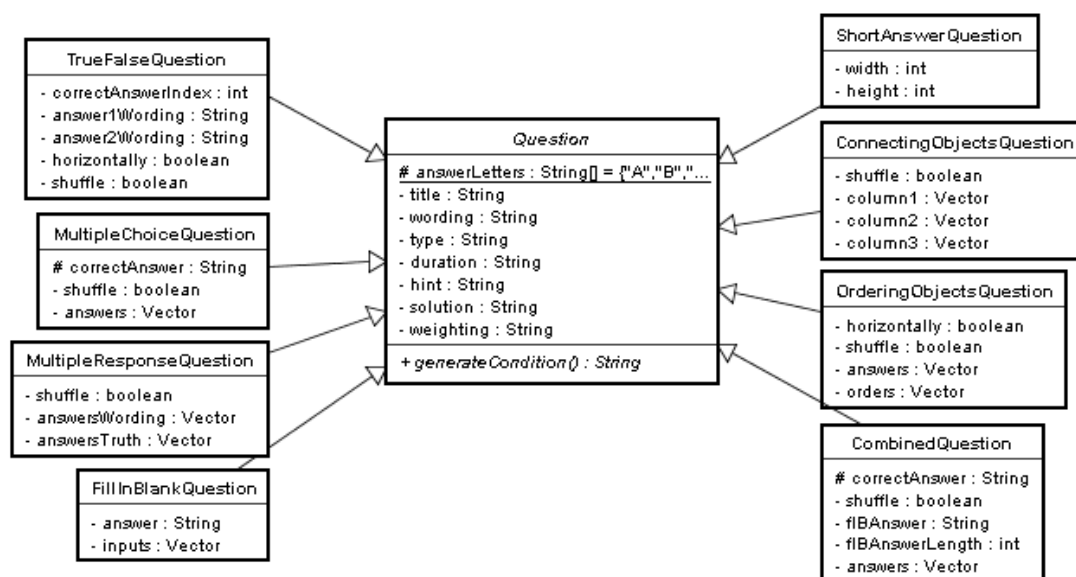
- `type` - typ uspořádání položek - náhodné nebo sekvenční
- `countOfRandomItems` - počet náhodných položek, které se mají vybrat
- `topic` - téma, podle kterého se má vybírat

Téma se používá v posledních třech typech uspořádání a výběru. V XML je uloženo v meta-datech položky a v aplikaci jde vlastně o atribut *Název otázky*, který je povinný u každé otázky. Názvy otázek by se proto z tohoto důvodu měly volit obecně (např. Algebra, Analýza) aby pokrývaly vždy určitou množinu otázek.

5.2.4 Podporované typy otázek

Aplikace podporuje 8 druhů otázek, jež budou v této kapitole podrobně popsány; vybrané typy otázek by měly pokrývat veškeré potřeby uživatele. Počínaje standardními typy otázek True/False, Multiple Choice, Multiple Response, Fill-In-Blank a Short Answer, umí aplikace pracovat i se složitějšími Ordering Objects (určování pořadí objektů), Connecting Objects (přirazování objektů, které k sobě logicky patří) nebo Combined, což je otázka typu Multiple Choice s možností vložit vlastní odpověď.

Pro ukládání otázek a odpovědí používá aplikace třídy zobrazené na obrázku 5.5.



Obrázek 5.5: Diagram tříd s otázkami

Všech 8 typů otázek je potomky abstraktní třídy *Question*. Atributy této třídy slouží pro uchovávání hodnot nezávislých na druhu otázky. Dále pak obsahuje abstraktní metodu *generateCondition()*, která je implementována v potomcích (tedy u každé z 8 otázek) a zajišťuje vygenerování výrazu pro hodnocení na základě vyplněných hodnot správných odpovědí u příslušné otázky.

5.2.4.1 Otázka typu True/False

True/False otázka je nejjednodušším druhem otázky, uživatel má na výběr ze dvou možností, musí vybrat právě jednu z nich.

Je Paříž hlavním městem Francie?

- Ano
- Ne

Obrázek 5.6: Příklad otázky True/False

Otázka má tyto možnosti nastavení:

- **Název otázky** - Neboli Téma. Používá se při výběru a uspořádání otázek v testu a pro orientaci v seznamu otázek.
- **Znění otázky** - Samotné znění otázky, lze vkládat multimédia (více viz. 5.3.4)
- **Správná odpověď** - Zvolení správné odpovědi, buď A nebo B
- **Zarovnání odpovědí** - Vertikální nebo horizontální uspořádání odpovědí
- **Náhodné pořadí** - Pokud zaškrtnuto, odpovědi budou náhodně uspořádány

5.2.4.2 Otázka typu Multiple Choice

V otázce Multiple Choice musí uživatel vybrat právě jednu z několika zobrazených možností.

Jaké je hlavní město **Francie**?

- Berlín
- Londýn
- Paříž

Obrázek 5.7: Příklad otázky Multiple Choice

Otázka má tyto možnosti nastavení:

- **Název otázky** - Neboli Téma. Používá se při výběru a uspořádání otázek v testu a pro orientaci v seznamu otázek.

- **Znění otázky** - Samotné znění otázky, lze vkládat multimédia (více viz. 5.3.4)
- **Odpovědi** - Znění odpovědí, lze vkládat multimédia (více viz. 5.3.4)
- **Správná odpověď** - Zvolení správné odpovědi
- **Náhodné pořadí** - Pokud zaškrtnuto, odpovědi budou náhodně uspořádány

5.2.4.3 Otázka typu Multiple Response

V otázce Multiple Response je zobrazeno několik možných odpovědí a uživatel musí označit ty správné. Pro N definovaných odpovědí je tak možný počet správných odpovědí v rozsahu 0 až N.

Které z následujících prvků jsou obsaženy ve vodě?

- Vodík
- Helium
- Uhlík
- Kyslík
- Dusík
- Chlór

Obrázek 5.8: Příklad otázky Multiple Response

Otázka má tyto možnosti nastavení:

- **Název otázky** - Neboli Téma. Používá se při výběru a uspořádání otázek v testu a pro orientaci v seznamu otázek.
- **Znění otázky** - Samotné znění otázky, lze vkládat multimédia (více viz. 5.3.4)
- **Odpovědi** - Znění odpovědí včetně jejich pravdivosti, lze vkládat multimédia (více viz. 5.3.4)
- **Náhodné pořadí** - Pokud zaškrtnuto, odpovědi budou náhodně uspořádány

Doplňte chybějící slova do textu Richarda III.

Now is the of our discontent made
glorious by these sons of .

Obrázek 5.9: Příklad otázky Fill-In-Blank

5.2.4.4 Otázka typu Fill-In-Blank

Neboli tzv. Cloze test. Jde o doplňování odpovědí do prázdných polí v textu.

Otázka má tyto možnosti nastavení:

- **Název otázky** - Neboli Téma. Používá se při výběru a uspořádání otázek v testu a pro orientaci v seznamu otázek.
- **Znění otázky** - Samotné znění otázky, lze vkládat multimédia (více viz. 5.3.4)

Při vkládání vstupního pole se volí jeho datový typ (řetězec, celočíselný nebo decimální) a maximální počet znaků.

5.2.4.5 Otázka typu Short Answer

Od uživatele se očekává vložení odpovědi vlastními slovy, přičemž odpověď může být libovolně dlouhá.

Popište jak startuje auto.



Obrázek 5.10: Příklad otázky Short Answer

Otázka má tyto možnosti nastavení:

- **Název otázky** - Neboli Téma. Používá se při výběru a uspořádání otázek v testu a pro orientaci v seznamu otázek.
- **Znění otázky** - Samotné znění otázky, lze vkládat multimédia (více viz. 5.3.4)
- **Odpověď** bude uživatel vkládat sám do textového okna při průběhu testu. Lze nastavit **velikost** tohoto **okna**, tj. počet znaků (šířka) a počet řádků (výška).

5.2.4.6 Otázka typu Ordering Objects

Od uživatele je očekáváno seřazení objektů podle jejich logického významu.

Uspořádejte dny v týdnu.

Úterý Neděle Čtvrtek Pondělí Sobota Pátek Středa

Obrázek 5.11: Příklad otázky Ordering Objects

Otázka má tyto možnosti nastavení:

- **Název otázky** - Neboli Téma. Používá se při výběru a uspořádání otázek v testu a pro orientaci v seznamu otázek.
- **Znění otázky** - Samotné znění otázky, lze vkládat multimédia (více viz. 5.3.4)
- **Odpovědi** - Znění odpovědí včetně jejich pořadí, lze vkládat multimédia (více viz. 5.3.4)
- **Zarovnání odpovědí** - Vertikální nebo horizontální uspořádání odpovědí
- **Náhodné pořadí** - Pokud zaškrtnuto, odpovědi budou náhodně uspořádány

5.2.4.7 Otázka typu Connecting Objects

Uživatel musí přiřadit každou položky z jedné skupiny odpovídající položce z druhé skupiny.

Otázka má tyto možnosti nastavení:

- **Název otázky** - Neboli Téma. Používá se při výběru a uspořádání otázek v testu a pro orientaci v seznamu otázek.
- **Znění otázky** - Samotné znění otázky, lze vkládat multimédia (více viz. 5.3.4)

Určete která města patří do jakých států.

Holmenkollen	Norsko
Trondheim	Norsko
Göteborg	Norsko
Tampere	Norsko
Lillehammer	Norsko
	Norsko
	Švédsko
	Finsko

Obrázek 5.12: Příklad otázky Connecting Objects

- **Položky** - Znění položek včetně jejich propojení, lze vkládat multimédia (více viz. 5.3.4)
- **Náhodné pořadí** - Pokud zaškrtnuto, odpovědi budou náhodně uspořádány

5.2.4.8 Otázka typu Combined

Kombinovaná otázka je složená z Multiple Choice a Fill-In-Blank otázek. Jde o typ otázky, kdy je zobrazeno několik možností na výběr a navíc je uživateli umožněno vložit svou vlastní odpověď.

Jaké je hlavní město **Francie**?

Berlín

Londýn

Peking

jiné

Obrázek 5.13: Příklad otázky Combined

Otázka má tyto možnosti nastavení:

- **Název otázky** - Neboli Téma. Používá se při výběru a uspořádání otázek v testu a pro orientaci v seznamu otázek.
- **Znění otázky** - Samotné znění otázky, lze vkládat multimédia (více viz. 5.3.4)
- **Odpovědi** - Znění odpovědí, lze vkládat multimédia (více viz. 5.3.4)
- **Správná odpověď** - Zvolení správné odpovědi
- **Náhodné pořadí** - Pokud zaškrtnuto, odpovědi budou náhodně uspořádány

5.2.5 Algoritmy hodnocení

Specifikace „Outcomes Processing“ obsahuje popis, jakým způsobem vypočítat souhrnné skóre na úrovni testu či sekce. Výsledná skóre jsou založena nad sekcemi a/nebo položkami. IMS QTI podporuje vestavěné algoritmy nebo možnost definování vlastních algoritmů.

Aplikace podporuje následující vestavěné algoritmy:

Number Correct Počet správně zodpovězených otázek, které byly vybrány a zobrazeny uživateli. Nebere se ohled na to, zda se uživatel pokusil odpovědět

Number Correct Attempted Počet správně zodpovězených otázek s ohledem na ty, na které uživatel odpověděl, ne pouze zobrazené

Weighted Number Correct Vážený počet správně zodpovězených otázek, které byly vybrány a zobrazeny uživateli. Nebere se ohled na to, zda se uživatel pokusil odpovědět. Jednotlivé otázky mají přiřazenu váhu

Weighted Number Correct Attempted Vážený počet správně zodpovězených otázek s ohledem na ty, na které uživatel odpověděl, ne pouze zobrazené. Jednotlivé otázky mají přiřazenu váhu

Percent Correct Algoritmus je odvozen z „NumberCorrect“ a „NumberCorrectAttempted“ algoritmů, kde procento správných odpovědí je definováno jako „NumberCorrect“ / „NumberCorrectAttempted“, tedy počet správných odpovědí ku počtu těch, na které se uživatel pokusil odpovědět

Sum of Scores Součet skóre otázek, které byly vybrány a zobrazeny uživateli. Nebere se ohled na to, zda se uživatel pokusil odpovědět

Sum of Scores Attempted Součet skóre otázek s ohledem na ty, na které uživatel odpověděl, ne pouze zobrazené

Weighted Sum of Scores Vážený součet skóre otázek, které byly vybrány a zobrazeny uživateli. Nebere se ohled na to, zda se uživatel pokusil odpovědět. Jednotlivé otázky mají přiřazenu váhu

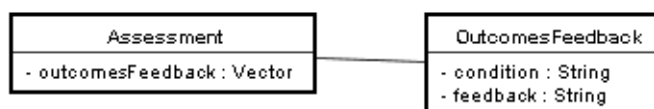
Weighted Sum of Scores Attempted Vážený součet skóre otázek s ohledem na ty, na které uživatel odpověděl, ne pouze zobrazené. Jednotlivé otázky mají přiřazenu váhu

Best K of N Pro výstup je použito nejlepších K skóre z množiny N zobrazených otázek. Nebere se ohled na to, zda se uživatel pokusil odpovědět

Specifikace IMS QTI verze 1.2 podporuje celkem 26 vestavěných algoritmů, přičemž u 10 z nich je pouze zmíněno, že budou implementovány v další verzi (Item Response Theory, Critical Item, Bayes net atd.). Pro aplikaci bylo vybráno deset základních nejpoužívanějších algoritmů, které by měly pokrýt potřeby uživatele.

Aplikace zajišťuje volbu jednoho z těchto algoritmů pro test a pokud je to třeba, nastavení jeho vlastností. Jedná se o přiřazení váhy (Weight) jednotlivým položkám testu (otázkám) nebo o nastavení počtu K nejlepších skóre u algoritmu Best K of N. API IMS zajistí uložení informací o zvoleném algoritmu a vlastnostech do XML souboru, včetně dalších potřebných parametrů pro umožnění výpočtu celkového skóre, které bude výstupem aplikace na průběh testů.

S algoritmy hodnocení úzce souvisí nastavení reakcí na celkovou hodnotu skóre v testu. Na obrázku 5.14 je třída `OutcomesFeedback`, která slouží k uchovávání těchto informací, a `Assessment`, která obsahuje seznam uživatelem vložených reakcí.



Obrázek 5.14: Vztah mezi třídami Assessment a OutcomesFeedback

Objekt `OutcomesFeedback` slouží pro uchování výrazu (`Condition`) a reakce (`Feedback`) na tento výraz. Výrazem může být např. „ $A \geq 0 \ \&\& \ A < 5$ “ a reakce na něj např. „Neuspěl jste“. IMS specifikace umožňuje použít pro vytvoření výrazu operátory `AND`, `OR`, `NOT`, přičemž umožňuje jejich libovolné zanořování. Aplikace pro jednoduchost umožňuje vytvoření jednoduchého výrazu („ $A > 0$ “) nebo složeného ze dvou jednoduchých výrazů („ $A > 5 \ \&\& \ A < 10$ “). V jednoduchém výrazu lze použít operátory `==`, `!=`, `>`, `<`, `>=`, `<=`. Při použití složeného výrazu má uživatel na výběr operátory `&&` a `||` mezi dvěma podvýrazy.

Aplikace neprovádí kontrolu zadaných výrazů, a to ani zdali jsou logicky správné (např. „ $A > 5 \ \&\& \ A < 3$ “), ani zdali jsou číselné hodnoty skóre použité ve výrazech vůbec při pozdějším průběhu testu dosažitelné. Při některých typech hodnotících algoritmů a jejich nastavení by totiž nebylo možné zjistit maximální dosažitelnou hodnotu skóre a tudíž u těchto výrazů nelze ověřovat jejich správnost. Veškerá zodpovědnost za správnost těchto podmínek tedy leží na uživateli.

5.3 Implementace

Cílem implementace je doplnit navrženou architekturu (kostru) aplikace o programový kód a vytvořit tak kompaktní systém [Vondrak02]. Tento krok byl realizován pomocí programovacího jazyka Java s využitím knihovny pro tvorbu grafického uživatelského rozhraní (GUI). Vzhledem k tomu, že množství vlastností aplikace již bylo popsáno v předchozí kapitole, budeme se v této věnovat pouze implementačním problémům. V následujících podkapitolách je podrobněji rozebrána struktura tříd a jsou popsány některé prvky implementace, které nejsou běžně součástí jednoduché GUI aplikace.

5.3.1 Struktura tříd

Aplikace je strukturována do balíků, jak ukazuje následující tabulka:

Balík	Popis
<code>cz.muni.fi.bronchus.gui</code>	třídy grafického uživatelského rozhraní
<code>cz.muni.fi.bronchus antlr.parsers</code>	třídy parseru vytvořené nástrojem ANTLR pro vyhodnocování výrazů
<code>cz.muni.fi.bronchus.bundles</code>	soubory pro lokalizaci
<code>cz.muni.fi.bronchus.resources</code>	obrázky, které aplikace používá
<code>cz.muni.fi.bronchus.questions</code>	vnitřní třídy pro ukládání objektů IMS

Tabulka 5.2: Uspořádání aplikace do balíků

5.3.2 Rozhraní Loader a Saver

Při práci s aplikací se neustále aktualizují objekty a jejich vlastnosti, v závislosti na akcích prováděných uživatelem. Necht' chce uživatel např. vytvořit test. Při jeho tvorbě se nejprve vytvoří objekt `Assessment`, nastaví se jeho vlastnosti, poté je do něj v závislosti na zvoleném druhu uspořádání a výběru přidán seznam otázek, které uživatel vytvořil, následuje další nastavování parametrů nad již vytvořenými objekty (např. nastavení typu algoritmů, nastavení vah jednotlivých otázek apod). Po závěrečné kontrole vytvořených objektů se vytvořený test uloží velice jednoduše použitím API IMS:

```
Saver saver = new Saver();
Map errorlist = saver.save(assessment, file, rewrite);
```

V případě neúspěšného uložení je vyvolána výjimka `SaveErrorException` a `errorlist` obsahuje seznam souborů, které nešly uložit spolu s výjimkami, které při ukládání příslušných souborů nastaly.

Podobně jednoduše probíhá načtení ze souboru:

```
Loader loader = new Loader();
Repository repository = loader.load(file);
```

Při načítání mohou nastat následující výjimky:

- `CreatedByOtherToolException` - pokud byl soubor vytvořen jinou aplikací
- `NonIMSXMLException` - pokud soubor není vystavěn podle specifikace IMS

Funkce importu API IMS funguje tak, že vrátí všechny otázky, které se daly načíst ze souboru:

```
Loader loader = new Loader();
List questions = loader.importQuestions(file);
```

Při importu je vrácena výjimka pouze pokud soubor není vystavěn dle specifikace IMS (`NonIMSXMLException`).

5.3.3 Zpracování odpovědí

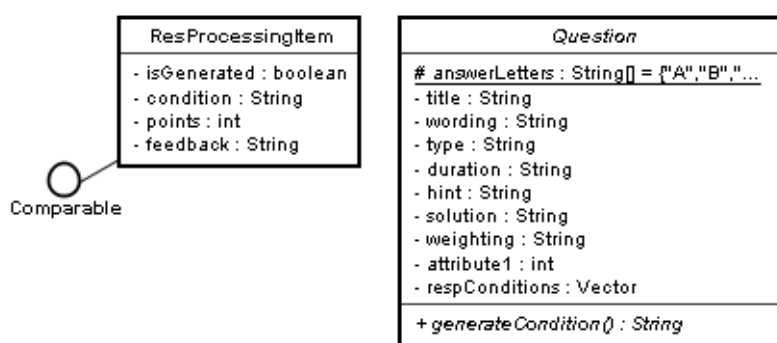
Při vytváření otázky je nutné vytvořit její hodnocení, neboli nějakým způsobem zaznamenat správné odpovědi, při průběhu testu přiřadit body za správnou odpověď a zobrazit případné reakce na uživatelem zadané odpovědi. Specifikace IMS obsahuje element `<resprocessing>` (zkratka pro `Response Processing`, tedy zpracování odpovědi), který poskytuje obsáhlé možnosti:

- lze vytvořit několik logických výrazů, které budou vyjadřovat uživatelské odpovědi při průběhu testu. Můžou tak být přiřazeny záporné body za zcela nesprávnou odpověď, nebo částečné bodové hodnocení za částečně správnou odpověď (typicky např. pro otázku typu `Multiple Response`)

- výrazy mohou být libovolně tvořeny pomocí operátoru AND, OR, NOT a operátorů ==, !=, >, <, >=, <=
- každému z těchto výrazů lze nastavit kromě bodů reakci, nápovědu a řešení

Aplikace implementuje plně všechny tyto možnosti. Pomocí nástroje ANTLR byla vytvořena gramatika, která ověřuje platnost uživatelem zadaných výrazů a tyto jsou pak předány spolu s reakcemi a body API IMS, které výrazy zpracuje a uloží do XML.

Ještě před popisem vlastní gramatiky věnujme pozornost uložení těchto prvků do objektů naší aplikace.



Obrázek 5.15: Diagramy tříd ResProcessingItem a Question

Každá otázka jakéhokoli typu rozšiřuje abstraktní třídu `Question`, která obsahuje seznam `respConditions`, jehož prvky jsou objekty třídy `ResProcessingItem`. Každý objekt této třídy má atributy pro uložení výrazu, bodů a reakce na tento výraz. Navíc tato třída implementuje rozhraní `Comparable` a definuje metodu `compareTo()`, která umožňuje při kontrolování zadaných výrazů v otázce seřadit je podle bodového hodnocení a zajistit tak správné bodové hodnoty (pokud to algoritmus hodnocení vyžaduje). Třída `Question` obsahuje také abstraktní metodu `generateCondition()`, která je implementována v každé z rozšiřujících tříd a slouží pro vygenerování výrazů ze zvolených odpovědí.

5.3.3.1 Gramatika pro tvorbu výrazů hodnocení odpovědí

Gramatika pro tvorbu výrazů hodnocení odpovědí je gramatika typu LL(2). Gramatika umožňuje:

- odkazovat se na odpovědi pomocí identifikátorů 'A' . . 'Z', které musí být vždy na levé straně výrazu
- relační operátory ==, !=, >, >=, <, <=
- operátor pořadí #

- logické operátory && (and), || (or) a ! (not)
- uzávorkování výrazů

Gramatika tak umožňuje plné pokrytí specifikace IMS elementu <respcondition>. Pro ilustraci uvádím několik příkladů výrazů:

- `A==1 && B!=3.1456 && C>-7.8e-4`
- `A==&hff || A==&o377 || A==&b11111111 || A==255`
- `A==#2 && B==#1`
- `A=="auto"`
- `A==true && B==false && C==true`
- `(A==1 || A==2) && (B>3.1 && B<3.2)`

Následuje výpis gramatiky, výrazy označené velkými písmeny jsou symboly, jejich význam je uveden v příloze.

```

expression
  : or_expression
  ;

or_expression
  : and_expression (OR and_expression)*
  ;

and_expression
  : relational_expression (AND relational_expression)*
  ;

relational_expression
  : LETTER (EQUAL|NOTEQUAL|LT|GT|LE|GE) constant
  | not_expression
  ;

not_expression
  : (NOT)? LPAREN! expression RPAREN!
  ;

constant
  : numeric_plusminus_constant
  | boolean_constant
  | string_constant
  | order_numeric_constant
  ;EQUAL

```

```

numeric_plusminus_constant
  : (PLUS|MINUS)? numeric_constant
  ;

order_numeric_constant
  : ORDER_INT_CONST
  ;

numeric_constant
  : INT_CONST
  | FLT_CONST
  | BINARY_INTEGER
  | OCTAL_INTEGER
  | HEXADECIMAL_INTEGER
  ;

boolean_constant
  : TRUE_LIT
  | FALSE_LIT
  ;

string_constant
  : STR_CONST
  ;

```

Jak bylo již řečeno, aplikace umožňuje vytváření výrazů pro hodnocení uživatelem zadaných odpovědí. Uplatnění této vlastnosti je zejména u otázek typu Multiple Response, kdy lze hodnotit studenta i za částečně zodpovězené otázky, a Fill-In-Blank, kdy vytvoříme podmínky pro správnost uživatelem zadané odpovědi. Mějme např. 3 odpovědi A, B, C, z nichž A je správně a B a C jsou špatně. Pak lze vytvořit např. následující podmínky:

- `A==true && B==false && C==false`

(všechny odpovědi dobře, 3 body)

- `(A==true && B==false && C==true) ||`
`(A==true && B==true && C==false) ||`
`(A==false && B==false && C==false)`

(2 odpovědi dobře, 2 body)

- `(A==true && B==true && C==true) ||`
`(A==false && B==false && C==true) ||`
`(A==false && B==true && C==false)`

(1 odpověď dobře, 1 bod)

- `A==false && B==true && C==true`

(všechny odpovědi špatně, -1 bod)

U otázky typu Fill-In-Blank musíme vytvořit podmínky pro hodnocení ručně, aplikace je negeneruje (podobně je to u otázek Short Answer a Combined v případě, že správná odpověď je typu Fil-In-Blank). Aplikace pro tyto případy obsahuje průvodce, který projde postupně všechny vstupní pole obsažené ve znění odpovědi a uživateli výrazně ulehčuje práci tím, že umožňuje vytvářet již správně strukturované podmínky. Obrázek tohoto průvodce lze najít v příloze.

5.3.4 Možnosti vkládání multimédií

Specifikace IMS QTI obsahuje pro každý prvek položky (znění otázky, odpověď apod.) element `<material>`, který slouží pro uložení obsahu znění tohoto prvku. Dle specifikace může tento element obsahovat elementy `<mattext>`, `<matemtext>`, `<matimage>`, `<matvideo>`, `<mataudio>`, `<matapplet>` a `<matapplication>`, jejichž význam je zřejmý. Díky nim lze umožnit, aby zobrazený obsah prvku obsahoval text, zvýrazněný text, obrázek atd. Druhou možností, jak by se mohly do obsahu, který chce uživatel zobrazit, přidat multimediální prvky, je použití HTML kódu, který poskytuje daleko větší možnosti formátování textu, jako třeba barevné zvýraznění textu, velikost písma, tabulky, seznamy atd. IMS QTI tuto druhou možnost také podporuje, neboť povoluje použití sekce CDATA (Character Data) ve svých elementech.

Při implementaci byla aplikována první z těchto metod, neboli použití předdefinovaných elementů IMS QTI. Pro přidávání multimediálního obsahu při vytváření otázek lze použít tlačítka na nástrojové liště.



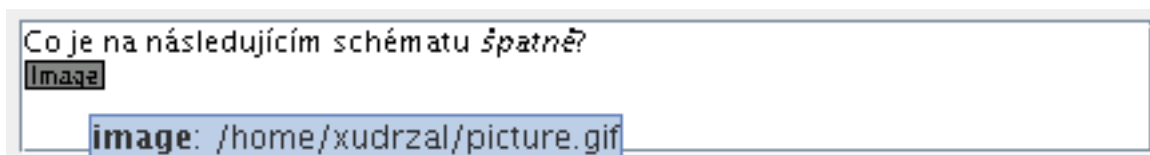
Obrázek 5.16: Tlačítka nástrojové lišty

Funkce těchto tlačítek je zřejmá:

- Italic - označený text se zvýrazní (kurzíva)
- Obrázek - vloží obrázek, podporované formáty jsou GIF, JPG, PNG
- Audio - vloží soubor s audio nahrávkou, podporované formáty jsou WAV, MP3
- Video - vloží soubor s videem, podporované formáty jsou MPEG, AVI
- TeX - vloží obrázek PostScript

Tlačítka lze použít pouze v některých polích aplikace. Jsou to pole s komponentou JTextPane, která podporuje zobrazení HTML kódu. Po vložení obrázku se objeví na místě kurzoru pouze

ikonka, tedy ve skutečnosti je vložen tag, který obsahuje informace o vloženém souboru (např. ``). Tyto pole jsou tedy při vytvoření objektu (otázky) uloženy jako HTML řetězce, které jsou pak zpracovány API IMS.



Obrázek 5.17: Příklad znění otázky

její HTML kód:

```
<html>
  <body>
    Co je na následujícím schématu <i>špatně</i>?
    
  </body>
</html>
```

a uložení v XML souboru:

```
<material>
  <mattext>Co je na následujícím schématu </mattext>
  <matemtext>špatně</matemtext>
  <mattext>?</mattext>
  <matimage imagetype="image/gif" uri="image/picture.gif" embedded=""/>
</material>
```

Komponenta JTextPane podporuje práci s HTML kódem, pokud se jí nastaví atribut „content-type“ na „text/html“. Poté umožňuje nejen zobrazení, ale i editování kódu za běhu. Bohužel nedosahuje kvalit webových prohlížečů, například vykazuje nepředvídatelné chování při práci se seznamem (UL). Z tohoto důvodu byla zvolena první metoda, tedy použití předdefinovaných elementů IMS QTI. Díky ní lze kontrolovat uživatelem vkládané prvky a tím pádem i validitu HTML kódu.

Při ukládání do souboru API IMS zajistí zkopírování odkázaných souborů s multimé-
dií do podadresáře k vytvářenému souboru, případně vrátí seznam souborů, které nešly zkopírovat.

5.3.5 Internacionalizace

Aplikace je internacionalizována do českého a anglického jazyka. Aplikace se dá spustit s parametrem `-locale <ISO Language Code>`, kde *ISO Language Code* je dvoupísmenný kód jazyka dle standardu ISO-639 psaný malými písmeny. Pro český jazyk je to *cs*, pro anglický *en*.

Pokud parametr není uveden, použije je jazyk uvedený ve vlastnostech operačního systému, na kterém je aplikace spuštěna.

O internacionalizaci se stará třída `MyResources`, která obsahuje metodu `getBundle`, která vrací lokalizovaný řetězec příslušného klíče.

```
public class MyResources {  
  
    private Locale locale;  
  
    public MyResources(Locale locale) {  
        this.locale = locale;  
    }  
  
    public String getBundle(String baseName, String key) {  
        return ResourceBundle.getBundle(baseName, locale).getString(key);  
    }  
}
```

Získaný řetězec pak můžeme použít:

```
jtitleLabel.setText(myResources.getBundle("TestsBundle","title_label"));
```

5.3.6 Vytvoření manuálu a jeho propojení s aplikací

Manuál k aplikaci byl vytvořen v XML Docbooku pomocí nástroje XML Mind XML Editor. Tento nástroj umožňuje vygenerovat XML jako HTML stránky, přičemž každá kapitola byla uložena do samostatného souboru. Vzhledem k tomu, že Swingové komponenty Java umožňují použití HTML, pak pro zobrazení manuálu stačilo pouze provést tyto kroky:

```
String source="/dokumentace/out/index.html";  
URL url = Tests.class.getResource(source);  
jTextPanel.setPage(url);
```

5.3.7 Java Web Start

Technologie Java Web Start je rozhraní mezi operačním systémem a Internetem, které umožňuje uživateli spouštět a ovládat aplikace umístěné na webu. S použitím návodu uvedeném v diplomové práci Lukáše Mikety [Miketa05] byla vytvořena verze aplikace, která je přizpůsobena běhu na webu a poskytuje tak uživateli alternativní možnost spouštění aplikace. Byly vytvořeny JNLP soubory, hlavní JAR soubor byly podepsány certifikátem. Pro uživatele to znamená, že po potvrzení důvěryhodnosti aplikace se spustí s neomezeným přístupem k lokálnímu systému souborů.

Kapitola 6

Závěr

Hlavním cílem této práce bylo vytvořit desktopovou aplikaci, která umožní uživateli pracovat s testy a ukládat je pomocí aplikačního rozhraní IMS QTI do formátu XML. Tato práce umožňuje kromě práce s repositáři otázek především tvorbu a úpravu testů, přičemž nabízí bohaté možnosti nastavení vlastností:

- 10 druhů vestavěných algoritmů hodnocení
- 8 druhů výběru a uspořádání položek
- 8 druhů otázek
- možnost nastavení reakcí, nápověd a řešení každé otázky
- možnost nastavení podmínek hodnocení každé otázky pro umožnění částečného hodnocení
- možnost nastavení každému objektu časovou prodlevu
- multimediální obsah otázek a odpovědí

Hlavní přínos této práce nespočívá ani tak ve vytvoření složitého grafického rozhraní v jazyce Java, jako spíše ve vytvoření mocného nástroje pro práci s testy dle standardu IMS QTI, který překonává většinu současných komerčních produktů.

Aplikace je vhodně navržena pro další možné rozšíření. Jedním z nich by mohlo být rozšíření formátovacích schopností otázek a odpovědí o editor, který by umožnil vkládat prvky HTML jako tabulky, seznamy apod. Struktura grafického rozhraní a styl průvodce umožňuje snadné rozšíření o další algoritmy hodnocení nebo druhy výběru a uspořádání položek.

Další pokračování práce na toto téma by mohlo implementovat Results Reporting XML standardu IMS, tedy způsoby zpracování výsledků. Navazovat na tuto práci bude ale zřejmě především systém pro testování studentů využívající aplikační rozhraní IMS a XML soubory vytvořené touto aplikací.

Literatura

- [CesnetES] Bažant, I., Drášil, P., Pitner, T., Šimák, B.: *Relevantní standardy v oblasti e-learningu*, Cesnet, 2004.
- [IMS] *IMS Question & Test Interoperability Specification*, IMS Global Learning Consortium, Inc., 2002, <http://www.imsglobal.org> <<http://www.imsglobal.org/question/index.html>>.
- [JDoc] *J2SE 1.5.0 API Specification*, Sun Microsystems, Inc., 2004, <http://java.sun.com> <<http://java.sun.com/j2se/1.5.0/docs/api/>>.
- [MacES] *Getting Started with eLearning Standards*, Macromedia, Inc., <http://www.macromedia.com> <<http://download.macromedia.com/pub/solutions/downloads/elearning/standards.pdf>>.
- [Miketa05] Miketa, L.: *Aplikace pro Java Web Start [diplomová práce]*, FI MU Brno, 2005. 5.3.7
- [SunES] *e-Learning Interoperability Standards*, Sun Microsystems, Inc., 2002, <http://www.sun.com/> <http://www.sun.com/products-n-solutions/edu/whitepapers/pdf/eLearning_Interoperability_Standards_wp.pdf>.
- [Tomola05] Tomola, M.: *API pro tvorbu otázek a testů [diplomová práce]*, FI MU Brno, 2002. 1
- [Vondrak02] Vondrák, I.: *Úvod do softwarového inženýrství*, VŠB Ostrava, 2005, <http://vondrak.cs.vsb.cz> <http://vondrak.cs.vsb.cz/download/Uvod_do_softwaroveho_inzenyrstvi.pdf>. 5.1, 5.2, 5.3

Rejstřík

ADL, 2, 3
AICC, 2, 3
AICC HACP, 3
ANSI, 2
ANTLR, 10, 28
ASI, 5, 12, 13
Assessment, 5, 6, 17, 27

Content Packaging, 3, 4, 7

DocBook, 34

Feedback, 6

Hint, 6

IEEE, 2, 3
IMS, 2–4, 11, 17, 35
IMS QTI, 3–7, 12, 13, 25, 32, 33, 35
IMS Results Reporting, 4
ISO, 2
Item, 5, 17

Java Web Start, 34

LMS, 1, 2
LOM, 3

meta-data, 3–5, 7, 17

Object Bank, 5, 6, 17
Outcomes Processing, 6, 25

položka, 5, 12, 26

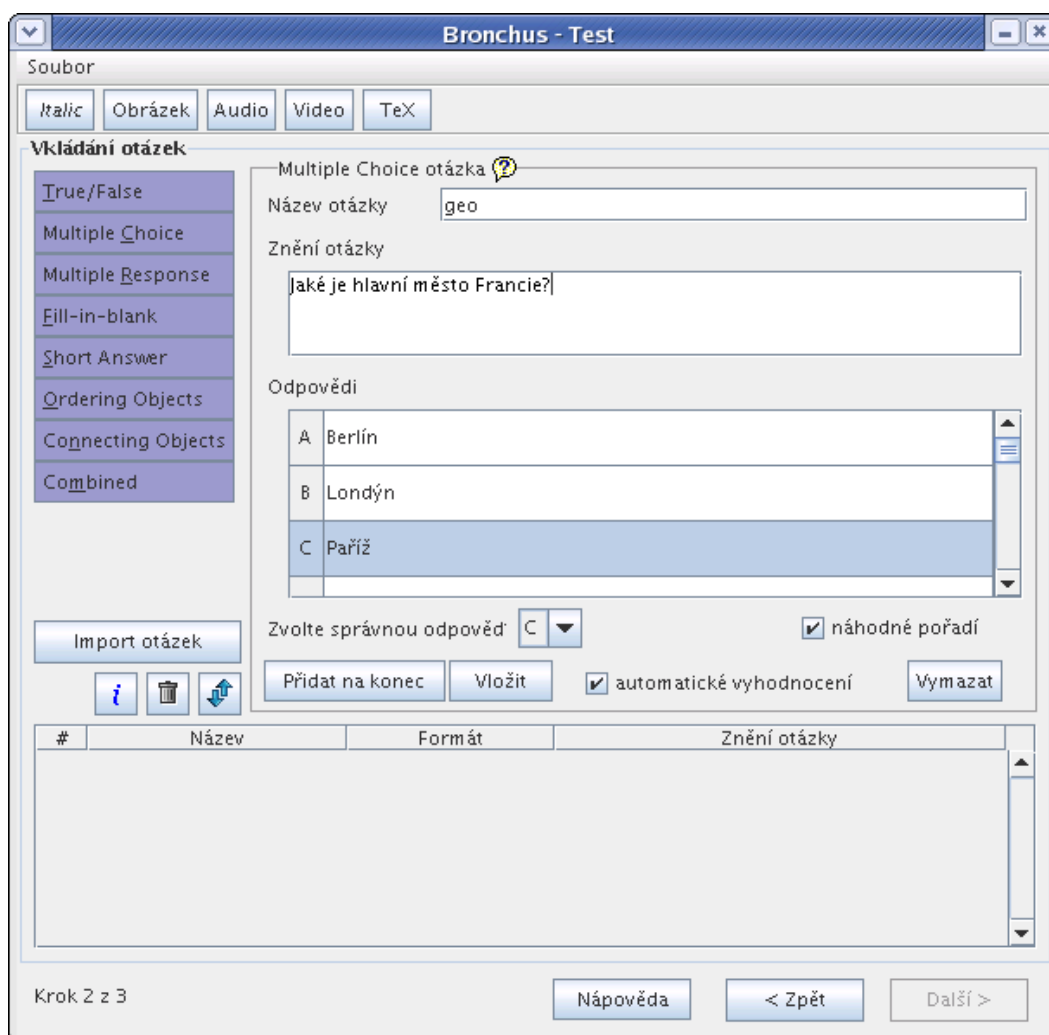
repositář otázek, 6, 14, 16
Response Processing, 6, 28
Results Reporting, 4, 35

SCORM, 2, 7
Section, 5, 17
sekce, 5, 12
Selection and Ordering, 7, 16
Solution, 6

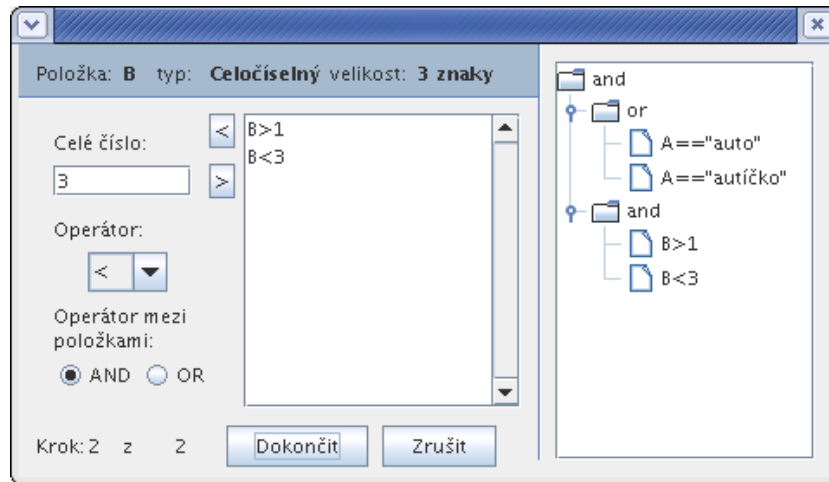
test, 6, 11–16, 26

Příloha A

Ukázka aplikace



Obrázek A.1: Vkládání otázek



Obrázek A.2: Průvodce pro vytváření podmínek hodnocení odpovědí

#	Název	Znění otázky	Fixní	Náhodně
1	zeměpis	Přiřadte hlavní města státům	<input checked="" type="checkbox"/>	
2	zeměpis	Jaké je hlavní město Mexika?	<input type="checkbox"/>	1
3	zeměpis	Jaká je hlavní surovina Egypta?	<input type="checkbox"/>	1
4	zeměpis	Kolik měří Mount Everest?	<input checked="" type="checkbox"/>	
5	zeměpis	Co je to vrstevnice?	<input type="checkbox"/>	2
6	zeměpis	Jaká je vzdálenost Země od Slunce?	<input type="checkbox"/>	2
7	zeměpis	Kdo byl první na Měsíci?	<input type="checkbox"/>	2
8	zeměpis	Na kolikáté rovnoběžce leží Česká Republika?	<input type="checkbox"/>	2
9	sport	Kdo letos vyhrál extraligu?	<input checked="" type="checkbox"/>	

Obrázek A.3: Příklad výběru a uspořádání typu Náhodných X + fixní pozice

Příloha B

Význam symbolů gramatiky

Symbol	Význam
LETTER	písmena A až Z
AND	&& logické and
OR	logické or
NOT	! logická negace
PLUS	+ znaménko plus
MINUS	- znaménko mínus
EQUAL	== je rovno
NOTEQUAL	!= není rovno
LE	<= menší nebo rovno
GE	>= větší nebo rovno
LT	< menší
GT	> větší
LPAREN	(levá závorka
RPAREN) pravá závorka
TRUE_LIT	true, pravdivostní hodnota pravda
FALSE_LIT	false, pravdivostní hodnota nepravda
STR_CONST	řetězcová konstanta ohraničená uvozovkami. Řetězce jsou automaticky považovány jako case-sensitive, tedy citlivé na velikost písmen.
INT_CONST	celočíslná konstanta
FLT_CONST	decimální konstanta (desetinná čárka se označuje tečkou)
BINARY_INTEGER	binární číslo "&b" (0 1)+
OCTAL_INTEGER	oktálové číslo "&o" ("0".."7")+
HEXADECIMAL_INTEGER	hexadecimální číslo "&h" ("0".."9" "a".."f")+
ORDER_INT_INTEGER	identifikátor pořadí "&" ("0".."26")

Tabulka B.1: Význam symbolů gramatiky

Příloha C

Obsah příloženého CD

Součástí této diplomové práce je CD, které obsahuje:

- `/text/` - zdrojový kód této práce ve formátu DocBook
- `/xudrzal.pdf` - tuto práci ve formátu PDF
- `/src/` - zdrojové kódy aplikace
- `/jnlp/` - soubory pro spuštění aplikace pro Java Web Start
- `/run/` - spustitelné soubory
- `/dokumentace/` - uživatelská dokumentace aplikace ve formátu DocBook
- `/dokumentace/html/` - uživatelská dokumentace aplikace ve formátu HTML
- `/javadoc/` - dokumentace tříd pro reprezentaci objektů

Příloha D

ANTLR software licence

We reserve no legal rights to the ANTLR—it is fully in the public domain. An individual or company may do whatever they wish with source code distributed with ANTLR or the code generated by ANTLR, including the incorporation of ANTLR, or its output, into commercial software.

We encourage users to develop software with ANTLR. However, we do ask that credit is given to us for developing ANTLR. By "credit", we mean that if you use ANTLR or incorporate any source code into one of your programs (commercial product, research project, or otherwise) that you acknowledge this fact somewhere in the documentation, research report, etc... If you like ANTLR and have developed a nice tool with the output, please mention that you developed it using ANTLR. In addition, we ask that the headers remain intact in our source code. As long as these guidelines are kept, we expect to continue enhancing this system and expect to make other tools available as they are completed.

Terence Parr

Příloha E

Dom4j software licence

Redistribution and use of this software and associated documentation ("Software"), with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain copyright statements and notices. Redistributions must also contain a copy of this document
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution
- The name "DOM4J" must not be used to endorse or promote products derived from this Software without prior written permission of MetaStuff, Ltd. For written permission, please contact dom4j-info@metastuff.com
- Products derived from this Software may not be called "DOM4J" nor may "DOM4J" appear in their names without prior written permission of MetaStuff, Ltd. DOM4J is a registered trademark of MetaStuff, Ltd
- Due credit should be given to the DOM4J Project - <http://www.dom4j.org>

THIS SOFTWARE IS PROVIDED BY METASTUFF, LTD. AND CONTRIBUTORS "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL METASTUFF, LTD. OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2001-2005 (C) MetaStuff, Ltd. All Rights Reserved.