

Vývoj a implementace webové aplikace s podporou notace IFML

Diplomová práce

Vedoucí práce:

doc. Ing. Ivana Rábová, Ph.D.

Bc. Jiří Syrový

Brno 2015

Na tomto místě bych rád poděkoval vedoucí mé diplomové práce doc. Ing. Ivaně Rábové, Ph.D. za obětavý přístup, cenné rady a připomínky, které mi poskytla během řešení této práce. Dále bych rád poděkoval svým rodičům za umožnění studia a své přítelkyni za oporu v průběhu celého studia.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Vývoj a implementace webové aplikace s podporou notace IFML**

vypracoval samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmetná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 18. května 2015

Abstract

Syrový, J. Development and implementation of web applications with the support of IFML notation. Diploma thesis. Brno: Mendel University in Brno, 2015.

This diploma thesis deals with the problems of development and implementation of web applications with the support of IFML notation. Processes in the sales department of middle-sized company which are focused on the electronic auctions are modelled through the use of UML 2.0 notation and BPMN 2.0 using a CASE software tool called Enterprise Architect. Then the complete process of development of web applications is made with the support of the product's life cycle using the modeling language IFML and a CASE software tool called WebRatio. The implementation of web applications is made using ASP.NET technology and Microsoft SQL Server.

Keywords

IFML, WebRatio, web application, ASP.NET, Microsoft SQL Server.

Abstrakt

Syrový, J. Vývoj a implementace webové aplikace s podporou notace IFML. Diplomová práce. Brno: Mendelova univerzita v Brně, 2015.

Tato diplomová práce se zabývá problematikou vývoje a implementace webové aplikace s podporou notace IFML. Procesy v obchodním oddělení středně velké firmy se zaměřením na elektronické aukce jsou namodelovány pomocí notace UML 2.0 a BPMN 2.0 za využití CASE nástroje Enterprise Architect. Poté je proveden kompletní proces vývoje webové aplikace s podporou životního cyklu produktu pomocí modelovacího jazyka IFML a CASE nástroje WebRatio. Implementace webové aplikace je provedena s využitím technologie ASP.NET a Microsoft SQL Server.

Klíčová slova

IFML, WebRatio, webová aplikace, ASP.NET, Microsoft SQL Server.

Obsah

1	Úvod a cíl práce	15
1.1	Úvod	15
1.2	Cíl práce.....	15
2	Teoretická východiska problematiky webových aplikací	17
2.1	Webové aplikace a technologie.....	17
2.1.1	Princip webových aplikací.....	17
2.2	Webové technologie.....	18
2.2.1	Skriptování na straně serveru	18
2.2.2	Skriptování na straně klienta.....	18
2.2.3	Kombinace strany serveru a klienta	18
2.2.4	ASP.NET	19
2.2.5	Microsoft SQL Server	21
2.3	Vývoj webových aplikací.....	21
2.4	Web Modeling Language (WebML).....	22
2.5	The Interaction Flow Modeling Language (IFML)	22
2.5.1	Specifikace požadavků	24
2.5.2	Doménový model	26
2.5.3	Front-end model.....	27
2.5.4	Model business logiky.....	30
2.5.5	Prezentační návrh.....	31
2.5.6	Realizace webové aplikace	31
2.5.7	Testování.....	31
2.5.8	Nasazení, údržba a rozvoj	32
2.6	Další možnosti modelování webových aplikací.....	32
2.6.1	Object-Oriented Hypermedia Design Model (OOHDM).....	32
2.6.2	Web Semantic Design Method (WSDM).....	33
2.6.3	UML-based Web Engineering (UWE)	34
2.7	CASE nástroj WebRatio.....	35

3	Metodická východiska práce	37
4	Vlastní práce	38
4.1	Charakteristika společnosti.....	38
4.1.1	Schéma organizační struktury společnosti	39
4.2	Procesy společnosti	39
4.2.1	Hlavní proces Obchod.....	41
4.3	Problémová doména a současný stav	43
4.4	Specifikace systémových požadavků na webovou aplikaci	44
4.4.1	Sběr požadavků.....	44
4.4.2	Analýza požadavků	46
4.5	Datová struktura.....	52
4.6	Tvorba IFML modelů.....	54
4.6.1	IFML model pro návštěvníka webové aplikace	55
4.6.2	IFML model pro vyhledovatele elektronické aukce.....	55
4.6.3	IFML model pro pozorovatele elektronické aukce.....	56
4.6.4	IFML model pro dodavatele.....	56
4.6.5	IFML model pro administrátora webové aplikace.....	56
4.7	Prezentační návrh	58
4.8	Implementace webové aplikace	59
4.9	Testování	61
4.10	Nasazení, údržba a rozvoj aplikace.....	61
5	Diskuze	62
6	Závěr	65
7	Literatura	67
A	Náhled jednotlivých částí webové aplikace	71
B	Seznam základních elementů IFML	73
C	Diagramy případů užití	76
D	IFML modely webové aplikace	79

Seznam obrázků

Obr. 1	Rozdělení technologií ASP.NET	20
Obr. 2	Proces vývoje webové aplikace pomocí notace IFML	23
Obr. 3	Ukázka formuláře se seznamem knih autora	28
Obr. 4	Ukázka IFML modelu	28
Obr. 5	Ukázka akce smazání produktů	30
Obr. 6	Schéma organizační struktury společnosti	39
Obr. 7	Hierarchická struktura procesů společnosti	40
Obr. 8	Průběh procesu Obchod	41
Obr. 9	Průběh subprocesu Nákup	42
Obr. 10	Diagram případů užití pro návštěvníka	48
Obr. 11	Diagram případů užití pro pozorovatele	49
Obr. 12	Datová struktura systému	54
Obr. 13	IFML model pro návštěvníka webové aplikace	55
Obr. 14	IFML model přihlášení uživatele	57
Obr. 15	Drátěný model přihlášení uživatele	57
Obr. 16	IFML model vytvoření elektronické aukce	58
Obr. 17	Drátěný model vytvoření elektronické aukce	58
Obr. 18	Přihlášení uživatele do webové aplikace	59
Obr. 19	Správa uživatelů	71
Obr. 20	Vytvořit novou aukci	72
Obr. 21	Diagram případů užití pro administrátora	76
Obr. 22	Diagram případů užití pro dodavatele	77

Obr. 23	Diagram případů užití pro vyhledávatele	78
Obr. 24	IFML model pro administrátora webové aplikace	79
Obr. 25	IFML model pro vyhledávatele elektronické aukce	80
Obr. 26	IFML model pro pozorovatele elektronické aukce	81
Obr. 27	IFML model pro dodavatele	82

Seznam tabulek

Tab. 1	Scénář případu užití: Registrace uživatele	50
Tab. 2	Scénář případu užití: Zobrazit vytvořené elektronické aukce	51
Tab. 3	Scénář případu užití: Upravit elektronickou aukci	52
Tab. 4	Seznam základních elementů IFML	73

1 Úvod a cíl práce

1.1 Úvod

Elektronické aukce slouží jako moderní nástroj pro optimalizaci nákladů v odběratelsko-dodavatelském řetězci. Elektronickou aukci si lze představit jako online výběr dodavatele ve sdílené webové aplikaci, kde na nejlepší nabídku dodavatele mohou ostatní dodavatelé reagovat zlepšením své nabídky. Takové aukce jsou vhodné pro snížení nákupní ceny s minimálním úsilím při vyjednávání a dále zajistí transparentnost nákupu. Jejich využití je široké, ať v oblasti nákupu či prodeje komodit, tak i v oblasti služeb. Elektronické aukce jsou používány jak v soukromém, tak veřejném sektoru. Použití vede ke snížení nákladů v oblasti strategického nákupu, který vybírá a hodnotí dodavatele včetně jejich cen.

V poslední době se webové aplikace stávají komplexními softwarovými produkty. Softwarové produkty, které bychom dříve našli instalované na klientském počítači, se nyní přenášejí do prostředí Internetu. Proto se nyní na Internetu můžeme setkat se systémy pro řízení obchodní firmy, správu zákazníků, správu elektronického bankovníctví či systémy pro organizaci týmu.

Kvalitních metodologií určených pro tvorbu softwaru nalezneme velké množství. Jsou to strukturované metodologie, objektové metodologie či univerzální jazyk UML (Unified Modeling Language) určený pro analýzu a návrh software. Avšak klasické metodologie určené pro tvorbu systému často neposkytují abstrakci pro specifikaci webových aplikací, které kladou důraz na hypertextové paradigma. Neposkytují notaci pro návrh navigace ve webové aplikaci, ani se nezaměřují na pohyb a orientaci uživatele mezi jednotlivými stránkami aplikace. Z tohoto důvodu vznikají metodologie pro vývoj webové aplikace. Mezi ty se řadí například metodologie OOHDM (Object-Oriented Hypermedia Design Method), WebML (Web Modeling Language) či IFML (The Interaction Flow Modeling Language). IFML není pouze sadou grafických specifikací, ale definuje i proces návrhu webové aplikace s podporou celého životního cyklu produktu.

1.2 Cíl práce

Brněnská společnost, se kterou je diplomová práce řešena ve spolupráci, se nachází před úpravou procesu nákupu v obchodním oddělení firmy. Toto oddělení se snaží optimalizovat náklady pomocí elektronické aukce.

Řešená problémová doména se týká problému nákupu komodit, tedy nákupu surovin a materiálu, pomocí elektronické aukce. Od této služby se očekává úspora času a efektivnost. Dále se očekává nákup komodit popřípadě služeb za skutečné tržní ceny a transparentnost nákupu.

Hlavním cílem této práce je vyvinout webovou aplikaci řešící problémovou doménu pomocí modelovacího jazyka IFML a CASE nástroje WebRatio.

K úspěšnému splnění hlavního cíle je nutné splnit i dílčí úkoly, mezi které se řadí:

- namodelovat a provést analýzu současných procesů v obchodním oddělení společnosti se zaměřením na elektronické aukce,
- vytvořit seznam funkčních a nefunkčních požadavků na webovou aplikaci a namodelovat je pomocí use case diagramu,
- namodelovat webovou aplikaci pomocí notace IFML a CASE nástroje WebRatio,
- na základě získaných poznatků realizovat výslednou webovou aplikaci, při její implementaci využít technologii ASP.NET.

2 Teoretická východiska problematiky webových aplikací

V části teoretická východiska problematiky webových aplikací bude nejdříve popsána architektura klient-server, komunikační protokol, objektově orientovaný jazyk ASP.NET s rozdělenými technologiemi a databázový server Microsoft SQL Server v podkapitole webové aplikace a technologie. Další podkapitola bude popisovat vývoj webových aplikací, a to pomocí metodologie Web Modeling Language (WebML), která je předchůdcem metodologie The Interaction Flow Modeling Language (IFML). Poté budou vysvětleny další možnosti modelování webových aplikací, a to zejména metodologie Object-Oriented Hypermedia Design Model (OOHDM), Web Semantic Design Method (WSDM) a UML-based Web Engineering (UWE). Na závěr bude popsán vývojový nástroj WebRatio.

2.1 Webové aplikace a technologie

V roce 1989 Tim Berners-Lee a Robert Cailliau v rámci svého působení v ženevském CERNu vytvořili službu WWW (World Wide Web). Vynálezci použili známý princip hypertextu, kdy soubory textů jsou navzájem propojeny pomocí odkazů. K tomu přidali komunikační protokol HTTP (HyperText Transfer Protocol). HTTP je internetový protokol, který je určen pro výměnu hypertextových dokumentů ve formátu HTML. Za dobu existence služby se web stal globální médiem s celosvětově propojenou sítí dokumentů.

2.1.1 Princip webových aplikací

Dokumenty uložené na webu lze najít ve formátu HTML (HyperText Markup Language). Tento hypertextový značkovací jazyk je používán pro tvorbu webových stránek, jejich propojení je určeno odkazy. HTML je vyvinuto z univerzálního značkovacího metajazyka SGML, který definoval HTML jako svoji podmnožinu (Casteleyn et al., 2009).

Získávání hypertextových dokumentů je založeno na komunikaci mezi klientem a serverem. Každý klient může pomocí webového prohlížeče zasílat žádosti o dokumenty jednomu či více serverů a servery mohou žádosti akceptovat, zpracovat je a vrátit klientovi požadované dokumenty. Komunikace nejčastěji probíhá přes HTTP připojení nebo jeho zabezpečenou podobu HTTPS (Casteleyn et al., 2009).

Pomocí jednotného identifikátoru zdroje, zkráceně URL (Uniform Resource Locator), lze přesně identifikovat hypertextový dokument v globální počítačové síti Internet. URL tedy určuje protokol použitý pro přenos zdroje, IP adresu či název stroje, na kterém je zdroj uložen. Dále URL určuje volitelné číslo portu, které vymezuje přístup k portu serveru, název dokumentu a jeho umístění (Brambilla et al., 2013).

2.2 Webové technologie

Komplexním systémům, mezi které jsou řazeny systémy pro řízení firmy, správu zákazníků či elektronické bankovníctví, nevyhovují statické webové stránky, které zobrazují stejný obsah stránky všem uživatelům nezávisle. Komplexní systémy vyžadují dynamické chování webových stran. K dynamickému chování webových stránek bylo vytvořeno skriptování a vytváření obsahu na straně serveru pomocí CGI, poté skriptování a vytváření obsahu na straně klienta pomocí JavaScriptu či jejich kombinace pomocí technologie AJAX. Tyto metody jsou popsány v textu níže podle autorů Casteleyn et al. (2009) a Haverbeke (2014).

2.2.1 Skriptování na straně serveru

Dle Casteleyn et al. (2009) je dynamické chování webových stránek například umožněno díky skriptům běžícím na straně serveru. Rozhraní CGI (Common Gateway Interface) slouží k propojení webového serveru a externí aplikace. Webový server předává požadavek klienta na tuto aplikaci a ta vrací výslednou prezentaci. Externí aplikace nejčastěji zpracovává skript a webovému serveru vrací statickou stránku, kterou server předá klientovi jako výstup.

Rozhraní CGI se používá od 90. let 20. století a byl to první způsob pro dynamické zpracování obsahu. Později ho nahradilo rozhraní FastCGI, které dokázalo zpracovat více požadavků a také skriptovací jazyky. Mezi nejčastější skriptovací jazyky pro skriptování na straně serveru patří: ASP, JSP, Perl, PHP, Python, Ruby a další.

2.2.2 Skriptování na straně klienta

Pro skriptování a vytváření obsahu na straně klienta se nejčastěji používá objekto-ově orientovaný skriptovací jazyk JavaScript. Byl představen v roce 1995, jeho autorem je Brendan Eich ze společnosti Netscape. JavaScript je od této doby přijat všemi hlavními webovými prohlížeči a lze jej používat při návrhu a tvorbě webových aplikací. Skripty se provádí ve webovém prohlížeči na straně klienta prostřednictvím reagování na události, které uživatel provedl z klávesnice nebo myši. Mezi vývojáři jsou často využívány knihovny, které jsou tvořeny s cílem ulehčit implementaci a usnadnit práci. Mezi nejčastěji používané knihovny lze zařadit: jQuery, Tojo Toolkit, MooTools či Prototype (Haverbeke, 2014).

2.2.3 Kombinace strany serveru a klienta

Spojením skriptovacího jazyka JavaScript a značkovacího jazyka XML (Extensible Markup Language) byl vyvinut nástroj AJAX (Asynchronous JavaScript and XML), který slouží pro tvorbu dynamických webových stránek. Mezi jeho pozitiva se řadí možnost tvorby interaktivních webových stránek, které umožní měnit obsah stránek bez nutnosti opětovného kompletního načtení stránky ze serveru. Použitím objektu XMLHttpRequest může skript vytvořit asynchronní požadavek na server. Tímto požadavkem žádá o určitá data. Je možné posílat i přijímat informace v různ

ných formátech, včetně formátu JSON, XML, HTML i textových souborů (Casteleyn et al., 2009) a (Haverbeke, 2014).

2.2.4 ASP.NET

Vývoj webové aplikace začal skriptovací platformou ASP (Active Server Pages) od firmy Microsoft v druhé polovině 90. let. Tato platforma byla určena pro tvorbu dynamických stránek založených na VB (Visual Basic) programovacím jazyku a jejich zpracování na straně serveru. Platforma ASP je předchůdce technologie ASP.NET, která je součástí .NET Frameworku pro tvorbu webových aplikací a služeb.

Před spuštěním se webová aplikace ASP.NET kompiluje a to pomocí dvou fází. V první fázi se kompiluje do jazyka MSIL (Microsoft Intermediate Language). MSIL je procesorově nezávislý jazyk, díky tomuto jazyku můžeme jednoduše přenášet kód mezi různými platformami. K druhé fázi dochází před tím, než se webová stránka spustí. Kód se v době provádění přeloží do nativního strojového kódu pomocí kompilátoru JIT (Just In Time). Technologie ASP.NET běží uvnitř prostředí CLR (Common Language Runtime). Použitím prostředí CLR vznikají výhody: automatická správa paměti, typová bezpečnost či podpora více vláken (multithreading) (Caison, 2002).

I když je teoreticky možné psát webové aplikace ASP.NET v jakémkoliv poznámkovém dokumentu, používají se komplexní vývojové nástroje. Jednou z možností usnadnění práce při vývoji webové aplikace je nástroj Visual Studio od firmy Microsoft. Výsledná webová aplikace je ukládána na serveru s verzí operačního systému Microsoft Windows Server a webovým serverem IIS (Internet Information Services), který vytvořila stejná společnost pro operační systém Windows.

Na obrázku (Obr. 1) lze vidět rozdělení jednotlivých technologií, které využívají stejné základní jádro ASP.NET. Mezi tyto frameworky patří Web Forms, Web Pages, Single Page Application a MVC.

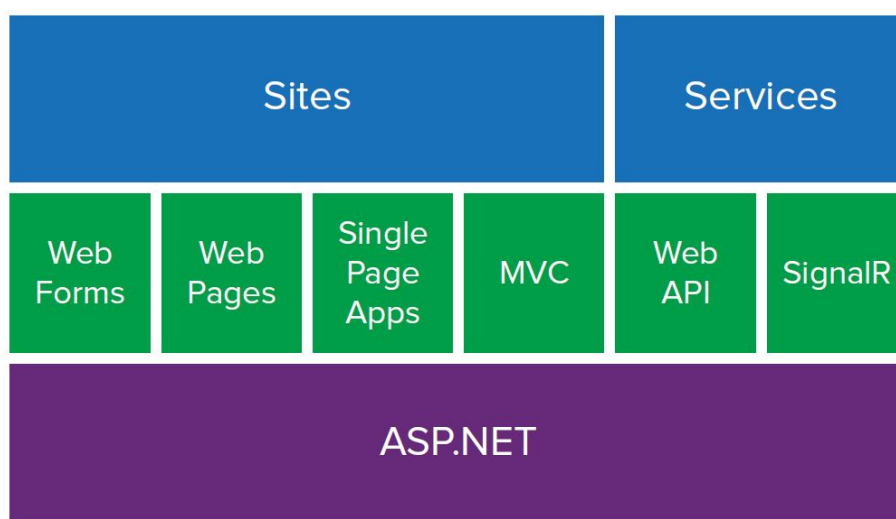
ASP.NET Web Forms byl vyvinut jako první framework z popisovaných technologií, a to v roce 2002. Web Forms odstraňoval problém bezstavovosti kombinací HTML, JavaScriptu a dvou technik. První z nich se nazývá View State, která uchovává informace o změnách hodnot vlastností ve skrytém formulářovém poli. Druhou z technik je Session State, která uchovává informace na serveru a předává pouze jednoznačný identifikátor (cookie nebo součást URL). Web Forms umožňuje rozdělit každou stránku do dvou souborů (oddělení prezentační a aplikační vrstvy). Soubor s příponou .aspx obsahuje HTML kód s vloženými značkami serverových komponent. Druhý soubor je code-behind, s koncovkou .vb či .cs podle programovacího jazyka VB.NET nebo C#. Tento soubor ošetřuje události komponent (Gaylord et al., 2013).

Framework ASP.NET MVC začal být vyvíjen v roce 2007, avšak nenahrazuje Web Forms, ale přichází jako alternativní technologie. MVC (Model-View-Controller) je softwarová architektura, která rozděluje webovou aplikaci do tří nezávislých komponent. První z nich je model, který obsahuje business logiku. Ta

zajišťuje validaci dat, provádí výpočty, dotazy a operace, které souvisí s daty. Druhou komponentou je pohled, který zajišťuje výstup uživatelského rozhraní. Poslední komponentou je kontroler, který komunikuje mezi komponentami model a pohled a také tyto komponenty propojuje (Gaylord et al., 2013).

Fink a Flatow (2014) ve své knize uvádí popis frameworku Single Page Application, zkráceně SPA. Tento framework navazuje na technologii ASP.NET MVC. Většina vývoje SPA se provádí na front-end, na rozdíl od tradičních webových aplikací, kde aplikační logiku zajišťuje server. SPA je webová aplikace, která používá pouze jednu HTML webovou stránku. Tato stránka se stáhne kompletně ze serveru a klient nemusí přejít na jinou stránku. Jsou v ní uloženy všechny podstránky, které se postupně skrývají a zobrazují podle práce uživatele na stránce. Data webové aplikace jsou ukládána lokálně. Změna dat se synchronizuje se serverem pomocí technologie AJAX. Tato technologie založená na JavaScriptu, kde celou aplikaci tvoří pouze jedna strana, se postupně prosazuje mezi technologiemi.

ASP.NET Web Pages je framework pro vývoj jednodušších webů, který je určen pro začínající vývojáře a programátory. Výrobce zmiňuje, že se jedná o alternativu k PHP či ASP. Pro dynamické webové stránky používá syntaxi Razor, která vytváří výsledný kód lépe čitelný. Oproti ostatním frameworkům nejčastěji používá vývojové prostředí WebMatrix, ale je možné použít i verzi Visual Web Developer. WebMatrix zahrnuje webový server IIS Express a databázi SQL Server Compact, tudíž je k dispozici kompletní nástroj pro vývoj webu (Brind, Spaanjaars, 2011).



Obr. 1 Rozdělení technologií ASP.NET
Zdroj: Gaylord et al., 2013.

Závěrem kapitoly lze zmínit, že ASP.NET, která je součástí .NET Frameworku od společnosti Microsoft, vydáním některého z novějších frameworků nepozastavila vývoj staršího frameworku. Z tohoto důvodu je na vývojáři výběr vhodné technologie, kterou pro vývoj aplikace zvolí.

2.2.5 Microsoft SQL Server

Microsoft SQL Server je výkonný relační databázový nástroj, který se používá pro ukládání dat v podnicích různých velikostí. Poskytuje bohatou sadu integrovaných služeb, které umožní různou práci s daty, jako je dotazování, vyhledávání, synchronizace, reporting či analýza dat (Jayanty, 2011). První samostatná verze serveru byla společností vyvinuta v roce 1995 s označením Microsoft SQL Server 6.0. Poslední verze 12.0 byla vydána v roce 2014 s označením SQL Server 2014. Server byl vydán v několika edicích, které se liší ve využití od vývoje a spuštění webových a malých serverových aplikací až po špičková datová centra s náročnou databází.

Jádro SQL serveru tvoří několik částí. Mezi důležité části lze řadit Database Engine Services, která je základem SQL serveru. Dále Analysis Services, která poskytuje technologii OLAP (Online Analytical Processing) pro analytické zpracování dat a funkci dolování dat (Data Mining) pro získávání užitečných informací z dat. Mezi rozšíření patří jazyk T-SQL (Transact-SQL), který slouží pro práci s daty (Jayanty, 2011).

2.3 Vývoj webových aplikací

V době vzniku prvních webových sídel se vytvářely statické webové stránky spojené pomocí hypertextových odkazů. Při vývoji těchto webových sídel se používala analýza, návrh a proces vývoje webové aplikace v minimální míře. V poslední době se webové aplikace stávají komplexními softwarovými produkty. Produkty, které bychom dříve našli instalované na klientském počítači, se přenášejí do prostředí Internetu. Proto se nyní setkáváme na Internetu se systémy pro řízení firem, správu zákazníků, elektronického bankovníctví a jimi podobnými.

Při vytváření těchto komplexních systémů se používají metodologie pro webové aplikace. Takovéto metodologie podporují proces vývoje webové aplikace. Pomocí vhodně zvolené metodologie, správné analýzy a návrhu při tvorbě těchto aplikací, lze zajistit její bezproblémový chod. Odkládání termínu dokončení či nárůst nákladů při vývoji jsou nejčastěji uváděné zápory při nepoužití metodologie.

Metodologie pro tvorbu webových aplikací vznikly na základě metodologií pro tvorbu softwaru. Tyto klasické metodologie určené pro tvorbu softwaru však neposkytují abstrakci pro specifikaci webových aplikací, které kladou důraz na hypertextové paradigma. Inovací metodologií pro tvorbu webových aplikací je návrh navigace. Lze říci, že navigace v rámci webové aplikace patří do oblasti uživatelského rozhraní. Klasické metodologie se návrhem navigace nezabývají.

Mezi metodologie pro modelování a generování kódu podle Tůmy (2012) patří například Object-Oriented Hypermedia Design Method (OOHDM), která je jednou z nejlepších metodologií, které jsou k dispozici. Dále Web Semantic Design Method (WSDM), UML-based Web Engineering (UWE), která je založená na UML. Mezi novější metodologie se řadí Web Modeling Language (WebML) a její nástupce The Interaction Flow Modeling Language (IFML).

2.4 Web Modeling Language (WebML)

Web Modeling Language (WebML) je již třetí generací metodologie pro analýzu a návrh webových aplikací, která byla vytvořena v rámci projektu Esprit s názvem Web-Based Intelligent Information Infrastructures v roce 1998 na italské univerzitě Politecnico di Milano. Od roku 1999 se WebML používá pro návrh firemních webových aplikací ve společnostech jako je Microsoft, Cisco Systems či Acer. Na podzim roku 2001 se návrháři a vývojáři rozhodli založit společnost WebRatio s cílem vytvořit stejnojmenný CASE nástroj založený na WebML (Ceri et al., 2003).

Cílem notace WebML bylo podpořit návrh a tvorbu datově náročných webových aplikací. Do těchto aplikací lze zařadit webové stránky pro přístup a uchování velkého objemu strukturovaných dat, typicky uložených jako záznamy v databázovém systému. K dosažení cíle WebML bylo využito existujících konceptuálních datových modelů, na nichž navrhli originální notaci pro vyjádření navigace a hypertextového rozhraní (Brambilla et al., 2013).

WebML není jen jazykem (sadou formálních grafických specifikací), ale definuje kompletní proces návrhu webových aplikací a podporuje celý životní cyklus projektu tzv. WebML Development Process. Samotná metodologie se skládá z několika modelů a pravidel, které určují, jak jsou modely provázány. Celek je komplexním modelem webové aplikace, která pomocí CASE nástroje umožňuje generovat kostru výsledné aplikace (Zelenka, 2003).

2.5 The Interaction Flow Modeling Language (IFML)

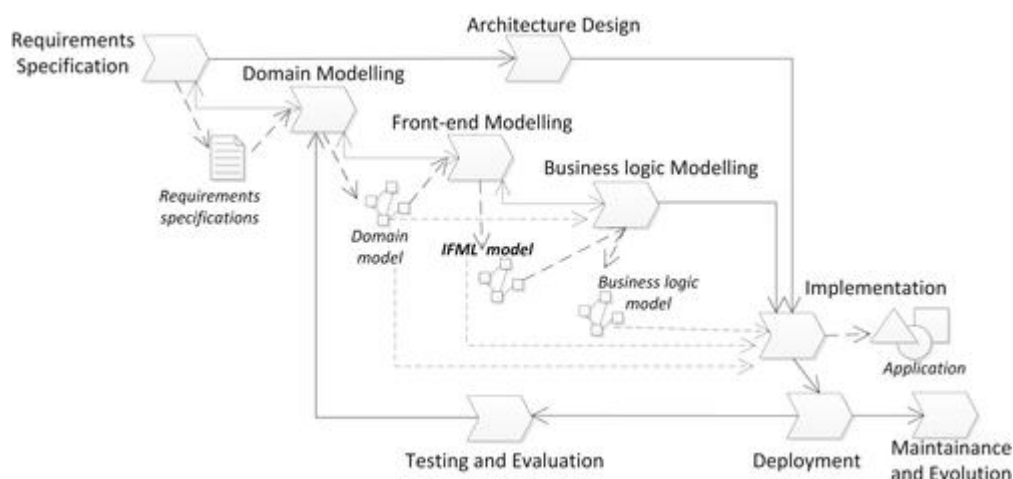
The Interaction Flow Modeling Language (IFML) je inspirován rozsáhlou desetiletou zkušeností s modelovacím jazykem WebML a vývojovým prostředím WebRatio. Je výsledkem výzkumné práce největší italské univerzity zaměřené na IT, která se jmenuje Politecnico di Milano. IFML byl přijat jako standard v březnu 2013 standardizační komisí Object Management Group (OMG). Během roku 2013 procházelo IFML úpravou díky sběru komentářů a poznámek. V únoru roku 2014 OMG vydala verzi beta 2, která se stala kandidátem na oficiální verzi IFML 1.0. V březnu téhož roku přijalo OMG specifikaci IFML 1.0 jako konečnou verzi. Únor roku 2015 byl stanoven jako datum uveřejnění této metodologie (IFML: The Interaction Flow Modeling Language, 2015).

IFML je standardizovaný modelovací jazyk, který se využívá v oblasti softwarového inženýrství. IFML obsahuje sadu grafických objektů, pomocí kterých se vytváří vizuální modely chování uživatelů s front-end systémy. Zaměřuje se na strukturu a chování aplikace, tak jak je vnímána koncovým uživatelem. Cílem IFML je poskytnout systémovým architektům, softwarovým inženýrům a vývojářům softwaru kvalitní metodologii, která by jim pomohla při vývoji systému. Cílem návrhu je možnost oddělit při modelování obsahovou část webu od jejich struktury a navigace nezávisle na prezentační části (Object Management Group, 2015).

Standart IFML je navržen pro vyjádření obsahu, interakci s uživatelem a ke kontrole chování front-end webových aplikací, které patří do následující oblasti aplikací (Brambilla, Butti, 2013):

- tradiční webové aplikace, které jsou založeny na HTML a HTTP,
- bohaté internetové aplikace podpořené HTML 5 standardem,
- mobilní aplikace,
- klient-server aplikace,
- desktopové aplikace,
- zabudované rozhraní člověk-stroj pro řízení aplikace,
- vícekanálové a kontextové aplikace.

IFML není jen sadou formálních grafických specifikací, ale definuje kompletní proces vývoje webové aplikace s podporou životního cyklu produktu (Brambilla, Butti, 2013).



Obr. 2 Proces vývoje webové aplikace pomocí notace IFML
Zdroj: Brambilla, Fraternali, 2014.

Dle autorů Brambilla a Fraternali (2014) je vývoj softwarových aplikací obvykle řízen pomocí agilních přístupů, které jsou založené na iterativním a inkrementálním vývoji. Každá iterace procesu vývoje vytváří prototyp neboli část verze systému. Tento životní cyklus je obzvláště vhodný pro webové a mobilní aplikace, které musí být rychle nasazeny a často upravovány během vývoje, aby se přizpůsobily požadavkům zadavatele. Na obrázku (Obr. 2) lze vidět kompletní proces vývoje webové aplikace pomocí notace IFML. Tento proces ve velké míře následuje klasický vývoj softwaru, který je dle metodiky Unified Process (UP) složen z pěti hlavních aktivit: požadavky, analýza, návrh, implementace a testování. Proces vývoje aplikace podle metodiky IFML se skládá z těchto částí:

- specifikace požadavků,

- doménový model,
- tvorba front-end modelu,
- model business logiky,
- návrh architektury aplikace,
- realizace webové aplikace,
- testování,
- nasazení, údržba a rozvoj.

2.5.1 Specifikace požadavků

Specifikace požadavků je důležitou součástí všech metodologií zabývajících se analýzou a návrhem systému. I z tohoto důvodu ji IFML definuje jako úvodní fázi procesu vývoje webové aplikace. Dle Zelenky (2004a) totiž na kvalitní specifikaci závisí úspěch systému. Protože jakákoliv nepatrná změna v zadání může například způsobit nedodržení termínu předání nebo zvýšit rozpočet systému.

Specifikací požadavků je činnost, při které analytik shromažďuje a formalizuje informace o problémové doméně a zjišťuje očekávané funkce. Vstupem je množina obchodních požadavků a veškeré dostupné technické, organizační a manažerské informace. Výstupem je specifikace požadavků, které slouží k vyhodnocení, zda zadání bylo splněno. Dle Ceri et al. (2003) se specifikace požadavků dělí na dvě fáze, a to na sběr požadavků a analýzu požadavků.

Sběr požadavků

Sběr požadavků se zaměřuje na získání obecného pohledu na aplikaci. Zdrojem informací jsou nejčastěji rozhovory se zadavatelem a dokumenty, které souvisí s řešeným problémem. Výstupem této části je poté soupis funkční specifikace. Dle Ceri et al. (2003) se má při sběru požadavků rozlišit několik oblastí:

- identifikace uživatelů – zde je cílem zjistit, kteří uživatelé budou webovou aplikaci využívat. Dále je vhodné rozdělit uživatele do skupin a zjistit základní charakteristiky každé skupiny.
- Funkční požadavky – řeší základní funkce, které by měla webová aplikace poskytovat svým uživatelům. Cílem funkčních požadavků je seznam určených funkcí, které jsou podporovány aplikací.
- Požadavky na data – v této oblasti by se měl analytik zaměřit na zjištění datových prvků, které se stanou základem pro návrh datové struktury.
- Požadavky na personalizaci – zde je nutné zjistit, zda obsah a služby budou pro všechny uživatele aplikace totožné, nebo se budou rozlišovat podle přístupových práv.
- Požadavky na specifické zařízení – sem patří problematika využití systému při použití specifických zařízení. Mezi takové zařízení lze řadit mobilní telefony, tablety apod.

- Nefunkční požadavky – tato oblast zahrnuje všechny ostatní požadavky, které jsou relevantní k dosažení obchodního cíle. Nefunkční požadavky pokrývají řadu klasických charakteristik, mezi které patří použitelnost, výkonnost, dostupnost, škálovatelnost, bezpečnost či rozšiřitelnost. Dále také programovací jazyk a technologie, kterou je aplikace implementována.

Analýza požadavků

Analýza požadavků formalizuje shromážděné poznatky z předchozí kapitoly sběr požadavků. Veškeré požadavky a získané materiály je nutné vyhodnotit a vytvořit z nich srozumitelné a přehledné výstupy. Výstupy této analýzy částečně kopírují sběr požadavků a budou popsány podle Zelenky (2004a) níže.

Skupiny uživatelů

V mnoha případech k webové aplikaci přistupují různí uživatelé. Proto je třeba uživatele identifikovat a rozřadit je do skupin. Každá skupina uživatelů by měla mít slovní popis, dále určeno, kteří uživatelé do ní patří a jak se systémem pracují. V některých případech lze rozlišit nadřizenou a podřizenou skupinu a určit hierarchii mezi skupinami.

Případy užití

Diagram užití (Use Case Diagram) a identifikace skupin uživatelů se navzájem doplňují. Diagram případů užití se používá k popisu chování vytvářeného systému z hlediska uživatele. Tento diagram také zachycuje skupiny uživatelů, které se systémem pracují. Dále určuje, jaké činnosti skupiny uživatelů vykonávají (Buchalceová, Pavlíčková, Pavlíček, 2007). Arlow a Neustadt (2007) ve své knize uvádí, že modelování případů užití zahrnuje následující aktivity: nalezení hranic systému, vyhledání aktérů, nalezení případů užití. Výstupem těchto aktivit je model případu užití se čtyřmi komponentami. Prvním z nich je hranice systému, což je ohraničení zobrazené kolem případů užití, neboli vyznačení oblasti modelovaného systému. Aktéři jsou role, které se přidělí uživatelům používajícím systém. Poté se nacházejí případy užití, což jsou činnosti, které mohou aktéři vykonávat. Relací se vytváří vztahy mezi aktéry a případy užití.

Pro specifikaci případů užití neexistuje přímo definovaný standard UML. Přesto Arlow a Neustadt (2007) uvádí šablonu tabulky pro každý případ užití, který obsahuje informace:

- název případu užití,
- jedinečný identifikátor případu užití (nejčastěji vyjádřeno číselně),
- stručný popis zachycující podstatu případu užití,
- aktéři zapojeni do případu užití,
- vstupní podmínky, které musí být splněny před spuštěním případu užití,
- hlavní scénář, který popisuje jednotlivé kroky,
- výstupní podmínky, které musí být splněny na konci případu užití,

- alternativní scénáře.

Datový slovník

Datový slovník vytváří seznam hlavních objektů (entit), které jsou identifikovány během požadavků. Datový slovník souvisí se strukturou dat při vytváření databáze.

Mapa webu

Při vytváření mapy webu je důležité znát vstupy, mezi které řadíme skupiny uživatelů, seznam případu užití a datový slovník. Výstupem činnosti je jedna nebo více map dle počtu skupin uživatelů, kde každá mapa webu je ve stručnosti popsána.

Grafický návrh aplikace

Grafický návrh aplikace je stanovení pravidel pro prezentaci webové aplikace, které mají být použity při návrhu prezentační vrstvy. U webových aplikací firem se bere v úvahu firemní styl (Corporate Identity), jehož součástí jsou barevná schémata, logotyp a grafické prvky.

Specifikace požadavků na webovou aplikaci nejsou omezeny pouze na metodologii IFML. Tuto fázi používají i jiné metodologie, například WebML, ze které IFML vychází. Jedná se o osvědčené postupy při vývoji webových aplikací.

2.5.2 Doménový model

Při vytváření doménového modelu vycházíme z fáze specifikace a sběru funkčních požadavků na webovou aplikaci. Kdy při konzultaci se zadavatelem lze identifikovat základní seznam entit. K těmto entitám lze později navrhnout jejich atributy a vztahy mezi entitami. Metodologie IFML nemá pevně definován modelovací jazyk, který se má při tvorbě doménového modelu použít. Mezi ně můžeme zařadit první metodu datového modelování pomocí entity-relationship modelování, která byla navržena v roce 1976 či modelovací jazyk UML. Při použití vývojového prostředí WebRatio se využívá konceptuálního modelování. Činnost doménového modelování se často nazývá také konceptuální modelování a v objektově orientovaném přístupu není mezi těmito modely rozdíl (Brambilla, Fraternali, 2014).

Konceptuální datový model definuje zobecnění oproti tvorbě datové struktury v konkrétním databázovém systému. Zobecnění znamená, že dostaneme nezávislý model na konkrétním systému a zároveň lze vygenerovat model do konkrétního implementačního prostředí. V případě použití nástroje WebRatio lze použít předdefinované datové typy blob, boolean, date, decimal, float, integer, password, string, text, time, timestamp a url. Datové typy jsou navrženy tak, aby byly nezávislé na konkrétním databázovém systému. Další možností je vytvoření vlastních datových typů. WebRatio každé vytvořené entitě přidá atribut OID (Object Identification), který identifikuje instance této entity. Při tvorbě relační databáze se bude atribut chovat jako primární klíč (Zelenka, 2004b).

Při návrhu datového modelu v nástroji WebRatio jsou povoleny pouze binární vazby (mezi dvěma entitami). Na vazbu se lze dívat jako na dvě vazby v opačných

směrech. Role poté představují pohled na vazbu ve směru od jedné entity k druhé entitě. Každá role má definovanou kardinalitu, a to minimální (hodnota 0 nebo 1) a maximální (hodnota 1 nebo N). V návrhu datového modelu nedefinujeme atributy cizí klíče z důvodu nezávislosti na konkrétním systému (Zelenka, 2004b).

Vytvořený datový model je výstupem modelování doménové struktury a je vstupem při vytváření IFML modelu.

2.5.3 Front-end model

Cílem front-end modelování je vytvořit části webové aplikace, které jsou zobrazitelné návštěvníkům v internetovém prohlížeči. Nejedná se přímo o grafický vzhled aplikace, nýbrž o model kompozice, tj. strukturu webové aplikace a navigační model, který slouží k propojení těchto stránek. Přestože tyto dva modely odlišujeme a každý má jiný význam, při vytváření front-end modelu je zakreslujeme společně.

Kompoziční model

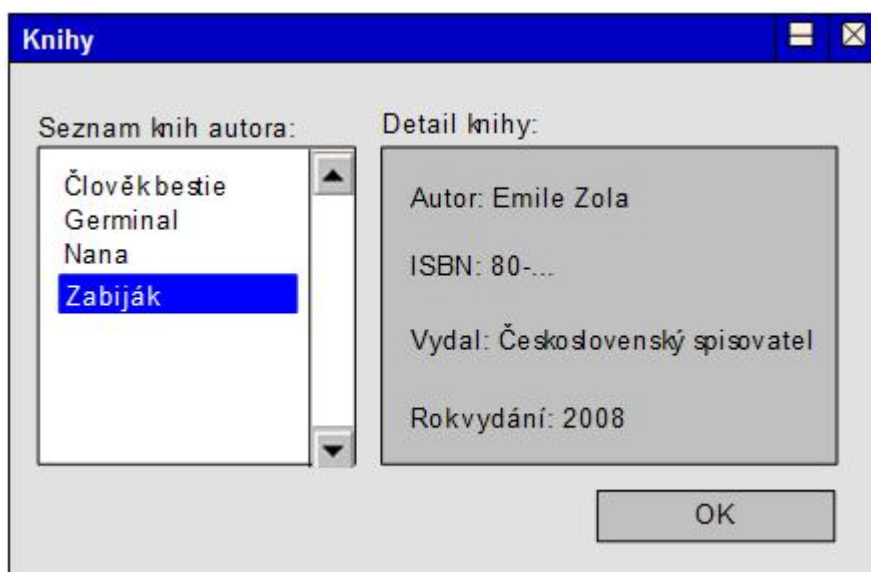
Cílem uspořádání webového rozhraní (kompozice modelu) je definovat strukturu webové aplikace, tedy její logickou strukturu.

IFML webovou aplikaci organizuje hierarchicky pomocí základních kontejnerů zvaných View Containers. Tyto základní prvky rozhraní zobrazují obsah, podporují interakci na stránce, obsahují jiné View Containers či komponenty pojmenované View Components. Jeden tento základní prvek lze definovat jako jednu webovou stránku aplikace, okno ve Windows či formulář v chytrém telefonu. IFML je totiž na platformě nezávislý jazyk. Kontejnery podporují navigaci, která je navázána na určitou událost. Pomocí navigačního toku můžeme procházet mezi kontejnery.

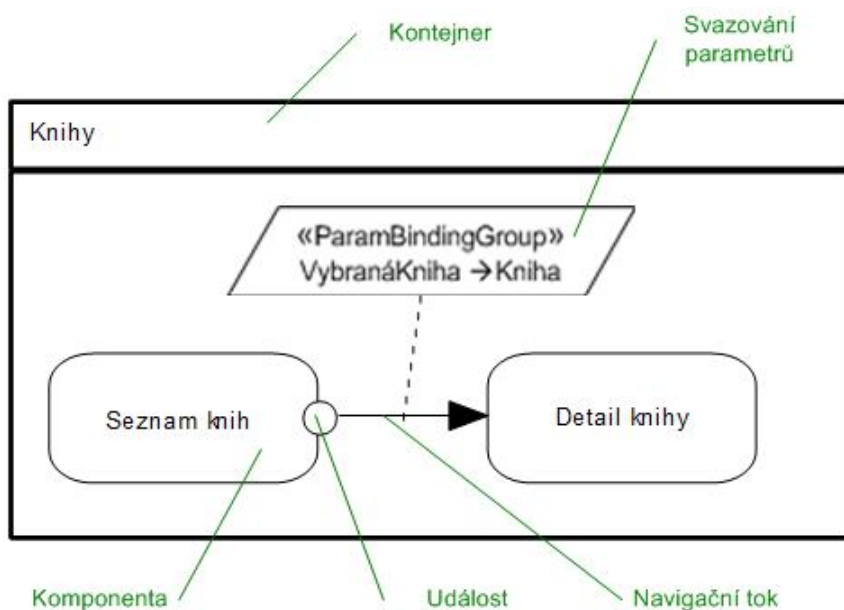
Komponenty jsou prvky, které zobrazují obsah nebo slouží pro zadávání dat (příkladem může být formulář, číslovaný seznam, zatrhávací tlačítko). Každá komponenta může mít vstupní a výstupní parametry. Komponenta může mít i vnitřní strukturu, která se skládá z jedné nebo více View Component Part. Tyto části komponenty jsou prvky rozhraní nebo vlastnosti, které nemohou existovat bez komponenty. Nejčastějšími částmi jsou prvky formulářů.

Kontejnery a komponenty mohou reagovat na události od uživatele, aplikace nebo vnějšího systému. Tím podporují interakci prvků s okolním světem. Samotná interakce je již závislá na konkrétní platformě, proto se pomocí IFML neprezentuje.

Na následujícím obrázku (Obr. 3) lze vidět formulář se seznamem knih vybraného spisovatele včetně detailu knihy ve vedlejším panelu. V případě, že je některá kniha vybrána, její detail se zobrazí v okně vedle. Na dalším obrázku (Obr. 4) pak lze vidět převedený formulář s interakcí zakreslený pomocí modelu IFML.



Obr. 3 Ukázka formuláře se seznamem knih autora
Zdroj: Rydval, 2013.



Obr. 4 Ukázka IFML modelu
Zdroj: Rydval, 2013.

Důsledek události se značí jako interakční tok (Interaction Flow), který je pomocí konektorů připojen na komponenty či kontejnery. Navigačním tokem lze vyjádřit změnu uživatelského prostředí. Událost může spustit akci, ta je spuštěna před změnou uživatelského prostředí. Příkladem je nutnost načíst data, než jsou uživateli zobrazena ve formuláři (Rydval, 2013).

Komponenty patří mezi základní prvky ve specifikaci IFML. Dle Wazlawicka (2014) existuje mnoho komponent, mezi nejpoužívanější patří:

- Details – zobrazení informací jedné instance.
- Multiple Details – zobrazení informací o kolekci instancí jedné entity.
- Simple List – představuje seznam instancí dané entity, umožňuje výběr jedné z instancí.
- List – jednoduchý seznam, který umožňuje posouvání a dynamické třídění.
- Checkable List – seznam instancí dané entity, který umožňuje vícenásobný výběr.
- Hierarchy – instance jsou uspořádané hierarchicky do stromové struktury.
- Scroller – slouží pro procházení seznamů instancí rozdělených do listů.
- Form – pole formuláře založené pro zadávání dat uživatelem.
- Multiple Form – formulář obsahující více polí pro zadání dat uživatelem.

Struktura webové aplikace

Rozsáhlé webové aplikace je nutné strukturovat do celků, proto se používá hierarchie stránek. Webové stránky (View Containers) se vyznačují některými rozlišovacími vlastnostmi, které zdůrazňují význam ve vytvářeném modelu (Brambilla, Fraternali 2014):

- implicitní stránka (D – Default View Container) – určuje výchozí stránku pro určitou oblast.
- Stále přístupná stránka (L – Landmark View Container) – určuje stránku, na kterou se lze dostat z jakékoliv jiné stránky. Pomocí těchto stránek lze zjednodušit navigační model. Stránky, které jsou takto označené, se používají při vytváření navigačního menu webové aplikace.
- Alternativní stránka (XOR – XOR View Container) – takto označená stránka v sobě zahrnuje vnořené stránky, na jejichž obsah se lze dostat z této stránky.

Navigační model

Kontejnery i komponenty podporují formu navigace. Navigace mezi těmito prvky určuje vzájemné propojení a tím se vytváří kompletní hypertextová struktura webové aplikace. K propojení se používají dva typy navigace (Brambilla, Fraternali, 2014):

- první je navigace obsahově nezávislá, která se přiřazuje na událost zdrojového prvku s cílem na koncový prvek. Při použití této navigace se nepřenáší informace mezi prvky (kontejnery a komponentami).
- Druhá navigace je obsahově závislá, která se také přiřazuje na událost zdrojového prvku s cílem na koncový prvek. Použitím obsahově závislé navigace naopak přenášíme informace mezi prvky. Takováto závislost mezi prvky se značí pomocí vazby parametrů (Parametr Binding).

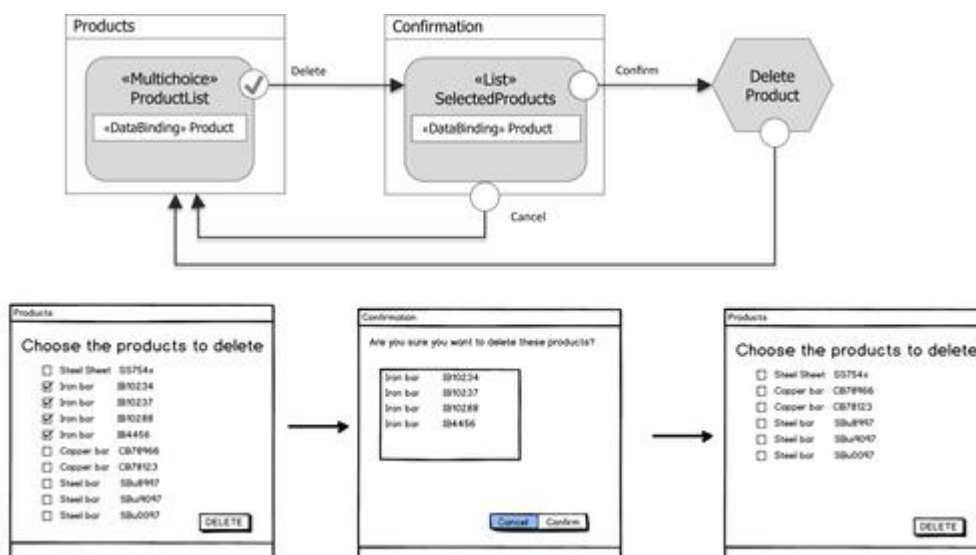
2.5.4 Model business logiky

Návrh business logiky umožňuje rozvinout a zpřesnit doménový model pomocí akcí, které specifikují vnitřní fungování webové aplikace. Mezi příklady akcí lze zařadit přidání, aktualizace a smazání dat. Jednotlivé akce se mohou nacházet na serveru nebo na straně klienta.

Nejčastější spuštění akce je pomocí události. Akce je poté spuštěna před samotnou změnou uživatelského prostředí. Příkladem je nutnost načíst data, než jsou uživateli zobrazena ve formuláři. Akce, které se modelují, jsou sdruženy s uživatelským rozhraním. Akce lze připojit ke komponentě s navigačním či datovým tokem pro předání parametrů mezi komponentami.

Definováním těchto akcí umožňuje WebRatio generovat a získat úplný kód webové aplikace.

Akce smazání vybraných produktů je znázorněna na následujícím obrázku (Obr. 5).



Obr. 5 Ukázka akce smazání produktů

Zdroj: Brambilla, Fraternali, 2014.

Autor Wazlawick (2014) ve své knize uvádí následující operace s daty:

- Create Operation – operace vytvoří novou instanci a záznam přidá do databáze.
- Delete Operation – využívá se k odstranění jedné nebo více instancí vybrané entity.
- Update Operation – operace se využívá k aktualizaci jednoho nebo více objektů stejné třídy.
- Connect Operation – operace slouží k vytvoření vztahu mezi instancemi.
- Disconnect Operation – tato operace odstraní vztah mezi dvěma instancemi.

Tyto operace mají speciální výstupní navigační toky. Existují dva typy výstupních odkazů pro operace popsané výše:

- OK tok – definuje, kam se aplikace přesměruje, když je operace úspěšně provedena pro každý objekt.
- KO tok – definuje přesměrování v aplikaci v případě, že operace selže u alespoň jednoho objektu.

Základní prvky modelovacího jazyka IFML jsou graficky znázorněné a popsané v příloze B (Seznam základních elementů IFML).

2.5.5 Prezentační návrh

Základním aspektem úspěšného uživatelského rozhraní je kvalitní grafický návrh, zvláště když je webová aplikace zaměřená na širokou veřejnost. Metodologie IFML záměrně ignoruje prezentační návrh a deleguje výsledný návrh prezentace na vývojový nástroj. CASE nástroj WebRatio vytvořené XML (eXtensible Markup Language) modely pomocí stylů XSL (eXtensible Stylesheet Language) převádí do výsledných naformátovaných webových stránek, které jsou kompatibilní s HTML5, CSS3 a JavaScriptem (Brambilla, Fraternali, 2014).

2.5.6 Realizace webové aplikace

Při implementaci lze využívat požadavky, datový model, IFML modely a veškeré návrhy vytvořené v přechozích krocích vývoje webové aplikace. V případě použití vývojového prostředí WebRatio lze vygenerovat plně funkční kód výsledné aplikace.

Fáze implementace není součástí IFML, ale vývojového prostředí WebRatio. Díky tomu, že jsou modely nezávislé na implementaci, lze k samotné realizaci použít programovací jazyky PHP, C#, VB.NET a další.

Vytvořené modely slouží jako vodítko při implementaci. WebRatio také podporuje generování vytvořené datové struktury na konkrétní databázový systém.

2.5.7 Testování

Testování patří mezi poslední fáze procesu vývoje webové aplikace pomocí notace IFML. Testování probíhá při vytvoření nové verze systému nebo lze testovat výslednou webovou aplikaci. Tato fáze ověřuje shodu aplikace na funkční a nefunkční požadavky, které jsou specifikovány v první fázi vývoje (Brambilla, Fraternali, 2014). Nejdůležitější body při testování jsou dle Jorgensena (2008) následující:

- funkční testování – ověřujeme chování aplikace s ohledem na funkční požadavky. Kontrolujeme, zda systém obsahuje funkce, které jsme obdrželi při komunikaci se zadavatelem.
- Testování použitelnosti – spočívá v testování vlastností aplikace, které nesouvisí s funkcemi aplikace, ale s ohledem na nefunkční požadavky.

- Výkonové testování – zde kontrolujeme dobu odezvy aplikace pod zátěží (větší počet současně pracujících uživatelů či větší objem dat), dále kontrolujeme i úroveň serveru, na kterém je webové aplikace nasazena.

2.5.8 Nasazení, údržba a rozvoj

Nasazení a poté údržba patří dle metodologie mezi závěr procesu vývoje webové aplikace. Nasazení je aktivita zprovoznění a zpřístupnění aplikace na vybraném serveru určeném zákazníkem. Mezi nasazení lze zahrnout vrstvu datovou, prezentační a funkční. Tato fáze procesu vývoje není metodologií IFML pevně stanovena, záleží na konkrétní aplikaci.

Údržba webové aplikace je součástí dohody mezi zadavatelem a dodavatelem webové aplikace. Jedná se o dohodu, zda zadavatel údržbu provádí samostatně nebo svěří údržbu dodavateli a případně v jakém časovém horizontu.

Rozvoj aplikace po jejím předchozím nasazení souvisí s potřebami zadavatele, které se v časovém období mohou změnit (Brambilla, Fraternali, 2014).

2.6 Další možnosti modelování webových aplikací

Kromě The Interaction Flow Modeling Language (IFML) a předchůdce Web Modeling Language (WebML) existuje řada dalších metod pro modelování a generování kódu webových aplikací. Mezi návrhy Web Engineering Models podle Tůmy (2012) patří například UML Web Application Extension (WAE), Object-Oriented Hypermedia Design Model (OOHDM), UML-based Web Engineering (UWE), Hypertext Design Model (HDM), Web Semantic Design Method (WSDM).

Cílem kapitoly však není podrobně rozebrat a srovnávat jednotlivé metodologie, které lze využít při modelování webových aplikací. Pro srovnání lze uvést tři přístupy a to metodologii OOHDM, která je jedna z nejznámějších a nejlepších, co je v současnosti k dispozici. Dále bude řešena metodologie WSDM a metodologie UWE, která je založena na UML.

2.6.1 Object-Oriented Hypermedia Design Model (OOHDM)

Metodologie Object-Oriented Hypermedia Design Model (OOHDM) se používá při návrhu webových aplikací. Jedná se o objektově orientovanou metodologii původně navrženou pro návrh hypertextových aplikací. Tuto metodologii vytvořili autoři Daniel Schwabe a Gustavo Rossi z univerzity PUC-Rio (Rio de Janeiro) v Brazílii.

Důvodem vzniku této metodologie byl dle autorů problém, že tradiční metodologie softwarového inženýrství neposkytují abstrakci při specifikaci webové aplikace, které využívají hypertext. Neposkytují zápis pro návrh a implementaci navigace ve webové aplikaci, kde navigace bývá nejdůležitější. A to z toho důvodu, že uživatel musí znát, kam směřuje a jak má dosáhnout cíle (Molhanec, 2001).

Proces vývoje podle této metodologie se skládá ze čtyř částí, které podporují inkrementální a iterativní vývoj. Konkrétně to jsou části: konceptuální modelování, návrh navigace, návrh abstraktního rozhraní a implementace. Těmto částem před-

chází sběr požadavků (Schwabe, 2008). Molhanec (2001) a Schwabe (2008) popisují tyto části následovně:

- sběr požadavků – prvním krokem je shromáždit požadavky na webovou aplikaci. Pro tento krok je důležité nejprve určit uživatele a úlohy, které budou vykonávat. Dále se vytváří scénáře pro každou úlohu a typ uživatele.
- Konceptuální modelování – v této fázi je cílem vytvořit konceptuální schéma, které reprezentují třídy, vztahy a subsystémy. Konceptuální schéma se skládá z množiny tříd, které jsou spojeny vztahy. Objekty jsou instancemi tříd. Třídy se v navigaci používají pro odvození uzlů a vztahy pro odvození odkazů.
- Návrh navigace – tento návrh se označuje za nejdůležitější v celém procesu vývoje. Navigační model je tvořen jako pohled na předchozí konceptuální model. Umožňuje vytvoření různých navigačních modelů pro různé uživatele. Navigace je vyjádřena pomocí schématu navigačních tříd a navigačních kontextů. Navigační třídy mají předdefinované typy a to uzly, odkazy a přístupové struktury. Specifikace navigačních transformací popisuje dynamické chování aplikace. Navigační kontext je množina uzlů, odkazů, tříd kontextů a případně dalších vnořených navigačních kontextů.
- Návrh abstraktního rozhraní – po návrhu navigace přichází definice uživatelského rozhraní. Zde se definuje způsob, jakým se prvky navigace budou zobrazovat uživateli. Protože je navigace a rozhraní odděleno, je možné, aby se stejná navigace zobrazovala různým uživatelům rozdílně.
- Implementace – skládá se ze tří činností. Mezi ně patří mapování informačních položek objektově orientovaného modelu do relačního, dále implementace kontextu, který vyžaduje ukládání stavové informace, a implementace rozhraní, které se ve většině případů generuje dynamicky dle obsahu.

2.6.2 Web Semantic Design Method (WSDM)

Metodologii Web Semantic Design Method (WSDM) navrhli v roce 1998 autoři De Troyer a Leune. Oproti ostatním metodologiím, které převážně vychází z objektového modelování, vychází tato metodologie z analýzy chování uživatele. Tato filozofie návrhu znamená, že různé cílové skupiny návštěvníků a jejich požadavky jsou brány jako výchozí bod pro návrh a hlavní struktura webu je od toho odvozena (De Troyer, Casteleyn, Plessers, 2008).

Proces vývoje webové aplikace podle této metodologie se skládá ze čtyř fází. Autoři De Troyer, Casteleyn, Plessers (2008) a Molhanec (2003) fáze popisují následovně:

- specifikace uživatelů a účel webového systému – cílem této fáze je identifikovat účel aplikace a specifikovat uživatele. Specifikace uživatelů se zakládá na analýze chování potenciálních uživatelů. Tito uživatelé jsou identifikováni a klasifikováni podle svých požadavků. Současně se bere ohled i na jejich

předpokládanou navigaci. Výstupem této fáze je poslání, které se formuluje v přirozeném jazyce a popisuje účel, předmět a uživatele webové aplikace.

- Třídění uživatelů – tato fáze navazuje na předchozí fázi, kde byli specifikováni uživatelé, kteří budou mít přístup k webové aplikaci. Nyní se různé typy uživatelů řadí do tříd uživatelů. Uživatelé se stejnými funkčními požadavky se stanou členy stejné třídy (skupiny). Uživatelé s dodatečnými požadavky, než obsahuje daná skupina, tvoří její podtřídou. Tímto způsobem se vytváří hierarchie mezi skupinami.
- Konceptuální modelování – tato fáze se rozkládá na dvě dílčí fáze. První fází je objektové modelování a druhou fází je návrh navigace. V průběhu objektového modelování se vytváří uživatelský objektový model, který je konceptuálním objektovým modelem. Konceptuální objektový model znázorňuje pouze třídy, které jsou důležité pro uživatele. Každému uživateli se vytváří uživatelský objektový model, který je pojmenovaný jako perspektivní objektový model. V druhé fázi, návrh navigace, se pro každý perspektivní objektový model vytváří navigační cesta. Navigační cesta vyjadřuje pohyb uživatele mezi informacemi, které mu chceme poskytnout.
- Implementace – je poslední fází metodologie WSDM. Samotná realizace webové aplikace se může provádět samostatně pomocí programovacího jazyka na základě informací získaných v průběhu předchozích fází, nebo může být generována automaticky. Automatické generování výsledné aplikace probíhá z konceptuálního návrhu.

I v této metodologii se používají konceptuální diagramy, na rozdíl od ostatních konceptuálních metodologií klade WSDM větší důraz na analýzu požadavků uživatele.

2.6.3 UML-based Web Engineering (UWE)

Metodologii UML-based Web Engineering (UWE) navrhla v roce 2000 autorka Nora Koch na univerzitě Ludwig-Maximilians v Mnichově. Metodologie UWE pokrývá veškeré fáze procesu tvorby systému, od sběru požadavků až po implementaci produktu. UWE se vyznačuje tím, že při modelování používá jazyk UML.

Metodologie poskytuje návod pro systematický a postupný návrh systému. Modelování probíhá v iterativním a inkrementálním návrhovém procesu. Mezi jednotlivé fáze patří analýza požadavků, konceptuální návrh, návrh navigace a poslední fází je návrh prezentace. Podrobně popisuje jednotlivé fáze autorka (Koch, Kraus, Hennicker, 2001) následovně:

- analýza požadavků – cílem analýzy požadavků je najít funkční požadavky na webovou aplikaci a reprezentovat tyto požadavky jako model případů užití.
- Konceptuální návrh – cílem je vytvořit konceptuální model aplikace s přihlédnutím k požadavkům a modelům případů užití. Výsledný konceptuální model se tvoří jako class diagram v UML.

- Návrh navigace – zachycuje základní strukturu webové aplikace a vztahy mezi jednotlivými stránkami.
- Návrh prezentace – v této fázi se vytváří prezentační model aplikace.

První dvě fáze jsou známými fázemi téměř v každém návrhu webové aplikace. Za přínos metodologie se považuje fáze návrh navigace. Tato fáze obsahuje tvorbu dvou modelů. První model je model navigačního prostoru a druhý je model navigační struktury. Model navigačního prostoru zobrazuje stránky webové aplikace a jejich propojení pomocí hypertextových odkazů. Webová stránka je modelována jako navigační třída. Odkaz mezi jednotlivými stránkami se značí šipkou. Model navigačního prostoru lze chápat jako upravení konceptuálního návrhu, při kterém se dbá na podrobnější vymezení navigace mezi stránkami.

Návrh prezentace podporuje tvorbu založenou na modelu navigace a informacích shromážděných během analýzy požadavků. Návrh prezentace se skládá z jednotlivých pohledů. Každý pohled je zobrazen jednou navigační třídou a určuje, jaké funkce může uživatel na stránce provádět.

Kladem této metodologie je, že se snaží z navržených modelů prostřednictvím vývojového nástroje generovat výsledný kód webové aplikace. Přičemž je možné si zvolit programovací jazyk, do kterého se kód vygeneruje.

2.7 CASE nástroj WebRatio

CASE nástroj WebRatio je vývojové prostředí podporující notaci IFML. Tento nástroj byl vytvořen spolu s notací WebML v roce 2001 na italské univerzitě Politecnico di Milano. V současné době je k dispozici jeho sedmá verze vydaná v prosinci roku 2014.

Vývojový nástroj WebRatio lze použít pro vývoj webové aplikace i mobilní aplikace. Tento nástroj využívá IFML standard jako vizuální modelovací jazyk pro zefektivnění procesu vývoje webové aplikace. Další standard, který podporuje, je Business Process Model and Notation (BPMN). BPMN je soubor principů a pravidel, které se používají pro znázornění podnikových procesů pomocí procesních diagramů. Poslední možnost, kterou podporuje, je modelování UML diagramů (Brambilla, Butti, 2013).

WebRatio automaticky generuje rychlé, bezpečné, škálovatelné a robustní webové a mobilní aplikace z modelů IFML a BPMN. Vytvořené aplikace jsou kompatibilní s HTML5, CSS3 a JavaScriptem. Aplikace vytvořené pomocí standardu Java/JSP 2.0 mohou být nasazeny na libovolném aplikačním serveru (příkladem se uvádí aplikační servery Oracle WebLogic Server a IBM WebSphere Application Server). Mezi možnostmi je nasazení aplikace v cloudu. Dle výběru licence umožňuje nástroj týmovou práci. Všechny zdrojové soubory jsou poté sdílené prostřednictvím CVS nebo Subversion a je zajištěn systém pro správu a vytváření verzí zdrojového kódu (Brambilla, Butti, 2013).

Instalace nástroje WebRatio je jednoduchý proces. Pro získání instalátoru aplikace je nejprve nutné se zaregistrovat na webových stránkách společnosti. Po

úspěšné registraci se nabízí možnost si stáhnout instalátor ve 32 bitové nebo 64 bitové verzi pro operační systém Windows, Linux či Mac OS X. Po instalaci CASE nástroje a jeho prvotním spuštěním jste vyzváni k aktivaci produktu pomocí aktivčního kódu. Po aktivaci produktu a potvrzení e-mailové zprávy lze nástroj používat (Rapid mobile App and Web application development Platform, 2015).

Uživatelské rozhraní aplikace se rozděluje do čtyř částí: hlavní menu a nástrojová lišta, hlavní pracovní plocha, levý sloupec a ladící okno. Hlavní menu a nástrojová lišta jsou umístěny v horní části produktu, odkud je k dispozici nastavení, funkce a nástroje. Hlavní pracovní plocha je vektorový grafický editor pro tvorbu jednotlivých modelů s boční lištou nabízející použitelné grafické prvky IFML. Levý sloupec slouží pro procházení stromové struktury projektu a zobrazení jednotlivých částí. V ladícím okně můžeme zobrazit chybová hlášení a problémy (Zelenka, 2004c).

Výsledná vygenerovaná webová či mobilní aplikace má třívrstvou architekturu založenou na modelu MVC (Model-View-Controller). Pro potřeby nástroje WebRatio je však architektura upravena. Každá stránka se skládá ze čtyř částí: objekt akce, služba stránky, JSP šablona a mapování akcí. Objekt akce dostává požadavek na zobrazení stránky, zpracovává požadavek a volá službu stránky, které předává parametry. Po dokončení výpočtu informuje, že je obsah stránky připraven. Služba stránky tvoří aplikační logiku, pomocí JSP šablony se generuje výsledný HTML kód. Poslední částí je mapování akcí, kde webová aplikace obsahuje konfigurační soubor, pomocí kterého čte řadič potřebné informace (Zelenka, 2004c).

Mezi další možnost, kde můžeme vytvářet modely pomocí využití notace IFML, patří open source vývojový nástroj Eclipse. V tomto případě se za pomoci pluginu vývojové prostředí rozšíří o návrh IFML.

3 Metodická východiska práce

Diplomová práce se zabývá oblastí vývoje webových aplikací. Konkrétně se jedná o návrh a vývoj aplikací pomocí metodologie IFML (The Interaction Flow Modeling Language). Modelovací jazyk vychází ze svého předchůdce, metodologie WebML. IFML je metodologie mladá, přijatá standardizační komisí OMG v roce 2014, a proto zatím málo rozšířená. Alternativní metodologie pro návrh webových aplikací jsou i přes jejich mladší vytvoření nedostatečně prosazovány a využívány. Z důvodu komplexní a složité webové aplikace, která řeší problém z praxe, je nutná její analýza a návrh za pomoci metodologie IFML.

Oblast analýzy, návrhu a tvorby webových aplikací je rozsáhlá problematika. Nejprve bylo nutné se seznámit s principy, na kterých jsou webové aplikace založené. Následovaly možné technologie, které lze při tvorbě aplikací použít. Tyto oblasti jsou součástí teoretické části problematiky webových aplikací.

Důležitou oblastí teoretické části této diplomové práce bylo seznámení se s metodologií IFML, která podporuje celý životní cyklus webové aplikace. Informace získané při studování této metodologie, která je klíčová při tvorbě webové aplikace, budou poté využity při řešení problémové domény v části vlastní práce.

Pro získání kompletního přehledu v oblasti vývoje webových aplikací byly v následující kapitole nastíněny další metodologie. Cílem však nebylo metodologie podrobně rozebrat a jednotlivě srovnávat. Proto byly uvedeny a popsány tři přístupy OOHD, WSDM a UWE.

Při vytváření modelů pomocí metodologie IFML se používá vývojový nástroj WebRatio. Popis a seznámení s tímto CASE nástrojem byla poslední oblast v teoretické části.

Po nastudování všech oblastí v teoretické části lze přejít na vlastní část práce. Ve vlastní práci bude nejprve popsána charakteristika společnosti, se kterou je problémová doména řešená. Dále bude znázorněna na schématu organizační struktura společnosti, popsány procesy společnosti a její hlavní proces Obchod a subprocess Nákup.

Následně dojde ke zjištění stavu problémové domény a jejího současného stavu. Poté budou určeny systémové požadavky na webovou aplikaci. Analýza probíhá pomocí diagramů případů užití a scénářů případů užití. Pro tvorbu diagramů s využitím nástroje Enterprise Architect bude použit grafický jazyk UML. Následné části fáze vývoje využívají jazyk IFML. To obnáší tvorbu datového modelu, front-end modelu a business logiky. Front-end model a model business logiky se modeluje do výsledného modelu IFML.

Poté bude následovat samotná realizace webové aplikace s fází testování a nasazení. Webová aplikace bude součástí procesu nákupu v obchodním oddělení společnosti, která chce pomocí nástroje snížit ceny nakupovaných surovin, minimalizovat úsilí a zajistit transparentnost nákupu.

Na závěr dojde ke zhodnocení přínosu webové aplikace pro společnost a srovnání možností použití notace IFML vzhledem k diagramům UML.

4 Vlastní práce

V kapitole vlastní práce bude nejprve popsán proces nákupu surovin a materiálu pomocí elektronické aukce v obchodním oddělení společnosti. Poté bude popsán návrh webové aplikace pomocí notace IFML. V této kapitole se využije znalostí nabytých v části teoretická východiska problematiky webových aplikací.

4.1 Charakteristika společnosti

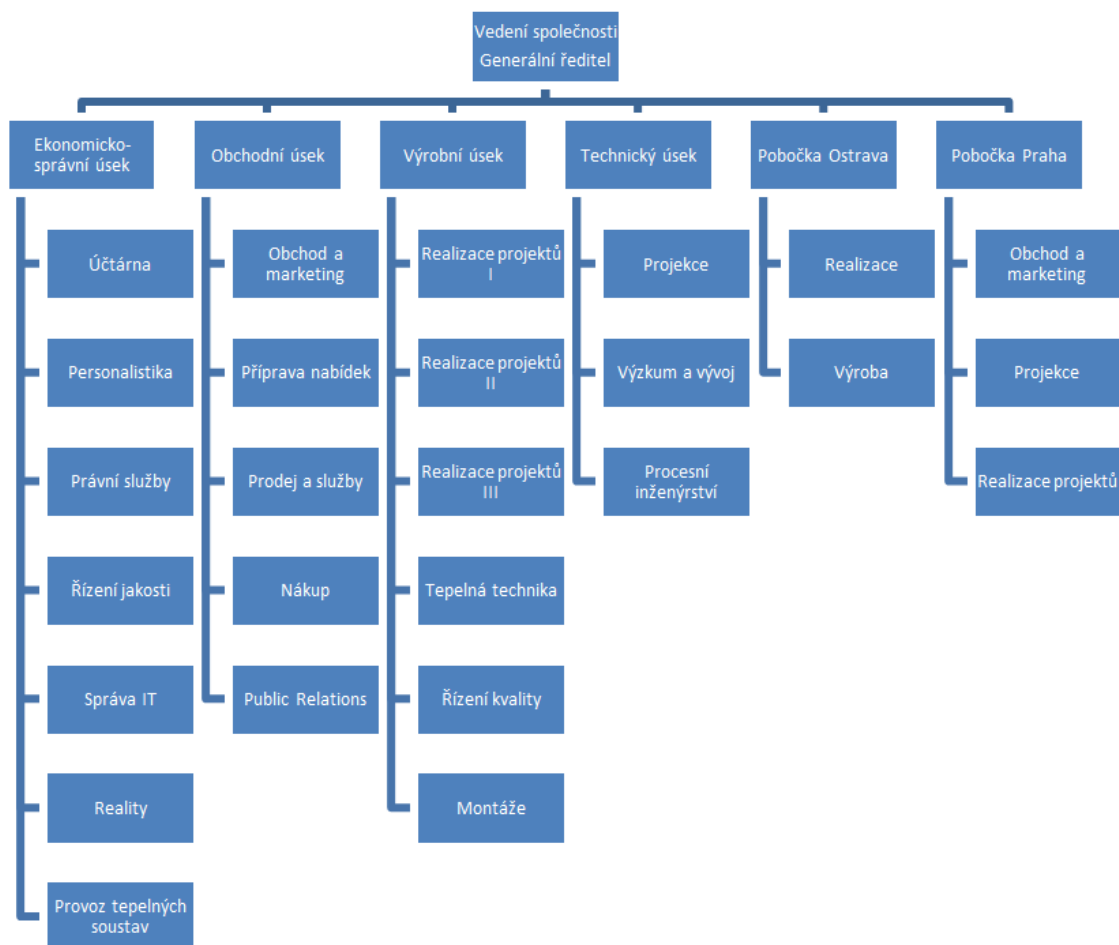
Brněnská společnost byla založena v roce 1993 jako společnost s ručením omezeným. V následujících letech zaznamenala rozvoj a vyprofilovala se jako dodavatel špičkových technologií a investiční akcí v oblasti energetiky a tepelné techniky. Společnost byla v roce 1999 převedena na společnost akciovou. V následujícím roce získala certifikát kvality ISO 9001. Od roku 2004 má společnost zaveden systém environmentálního managementu (EMS), který se zaměřuje na sledování a zlepšování činností podniku, které ovlivňují nebo mohou ovlivnit kvalitu životního prostředí.

Klíčovým prvkem ve výrobním programu společnosti je výstavba a rekonstrukce zdrojů pro výrobu tepla, elektrické energie a rozvodů tepla. V oboru nabízí společnost bohaté zkušenosti nabyté při výstavbách a rekonstrukcích tepláren či elektráren. Inženýrskou a dodavatelskou činností se zaměřuje na vytopny a bioplynové stanice s důrazem na výrobu energií z obnovitelných zdrojů.

Výstavba a rekonstrukce rozvodů tepla se realizuje během modernizace soustav centrálního zásobování teplem. Společnost je hlavním dodavatelem potrubních systémů a armatur zahraničních výrobců. Kompaktní předávací stanice z vlastní výroby, které jsou určeny k zásobování objektů teplem a teplou užitkovou vodou, doplňují nabídku tepelné techniky. V nabídce lze najít i provozování soustav zásobování teplem.

Mezi další obory činnosti společnosti se zahrnuje realizace pozemních staveb, vodohospodářských staveb a inženýrských sítí, zahrnujících stavbu kanalizací, vodovodů nebo čističek odpadních vod.

4.1.1 Schéma organizační struktury společnosti



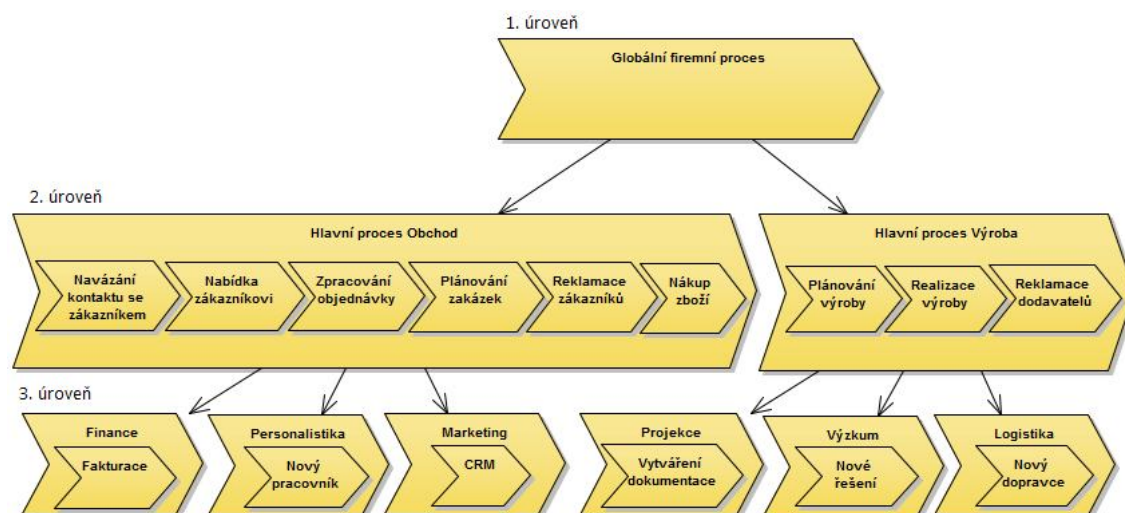
Obr. 6 Schéma organizační struktury společnosti

Na obrázku (Obr. 6) vidíme organizační strukturu společnosti. Ve vedení společnosti je generální ředitel. Společnost se rozděluje na čtyři úseky a dvě pobočky. Mezi úseky patří ekonomicko-správní, obchodní, výrobní a technický. Ty vedou ředitelé jednotlivých úseků. Pobočky nalezneme v Praze a Ostravě. Každý úsek se dále rozděluje na oddělení, za které odpovídá vedoucí oddělení. V této práci je důležitý obchodní úsek, který se stará o nákup surovin a materiálu pomocí elektronické aukce.

4.2 Procesy společnosti

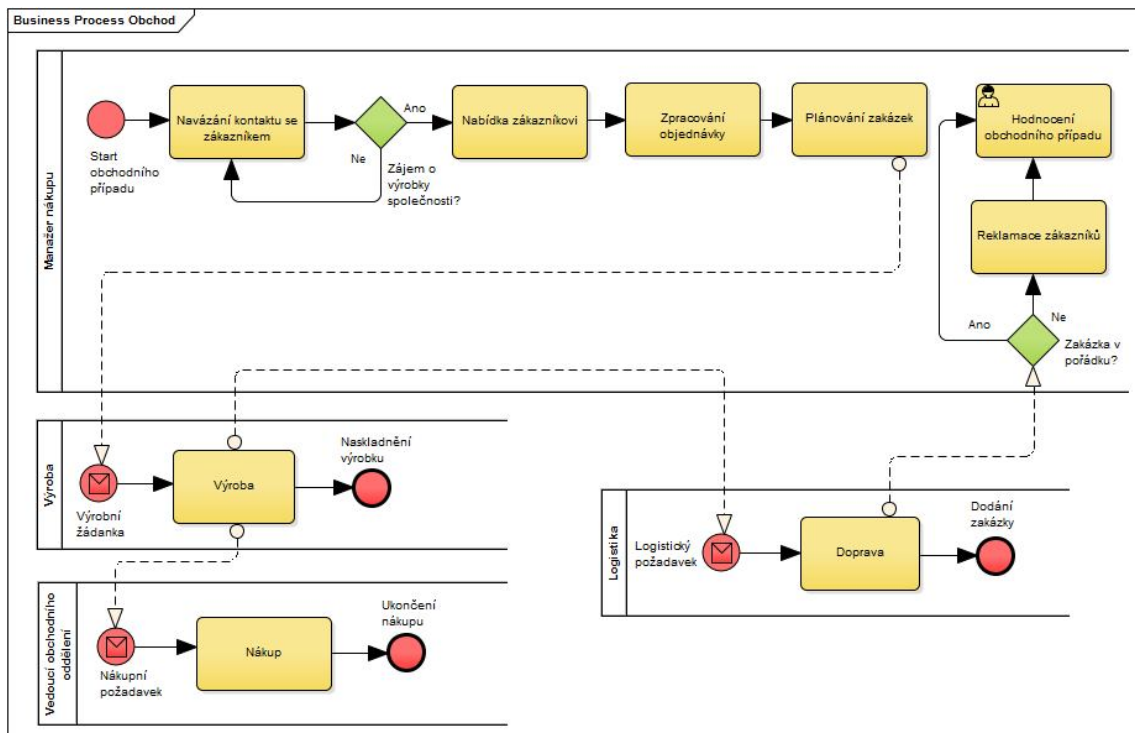
Úvod ve vlastní části práce je procesní modelování, kde je zvolen diagram pro zobrazení hierarchicky uspořádaných procesů společnosti. V tomto diagramu (Obr. 7) jsou jednotlivé procesy rozděleny do tří hierarchicky uspořádaných úrovní. Nejvyšší úroveň je globální proces společnosti. Podnikatelská činnost se zaměřuje na výrobu a rekonstrukci zdrojů pro výrobu tepla, elektrické energie a rozvodů tepla.

Z těchto důvodů jsou zvoleny mezi hlavní procesy Obchod a Výroba, které se nacházejí ve druhé úrovni struktury. Těmito procesy vzniká přidaná hodnota společnosti. Ve třetí úrovni se nacházejí podpůrné procesy jako Finance, Personalistika, Marketing, Projekce, Výzkum a Logistika, které slouží pro správné fungování procesů Obchod a Výroba. Ve společnosti lze najít i subpodpůrné procesy. V této práci je však nejdůležitější proces Obchod a její subproces Nákup zboží pomocí elektronické aukce.



Obr. 7 Hierarchická struktura procesů společnosti

4.2.1 Hlavní proces Obchod



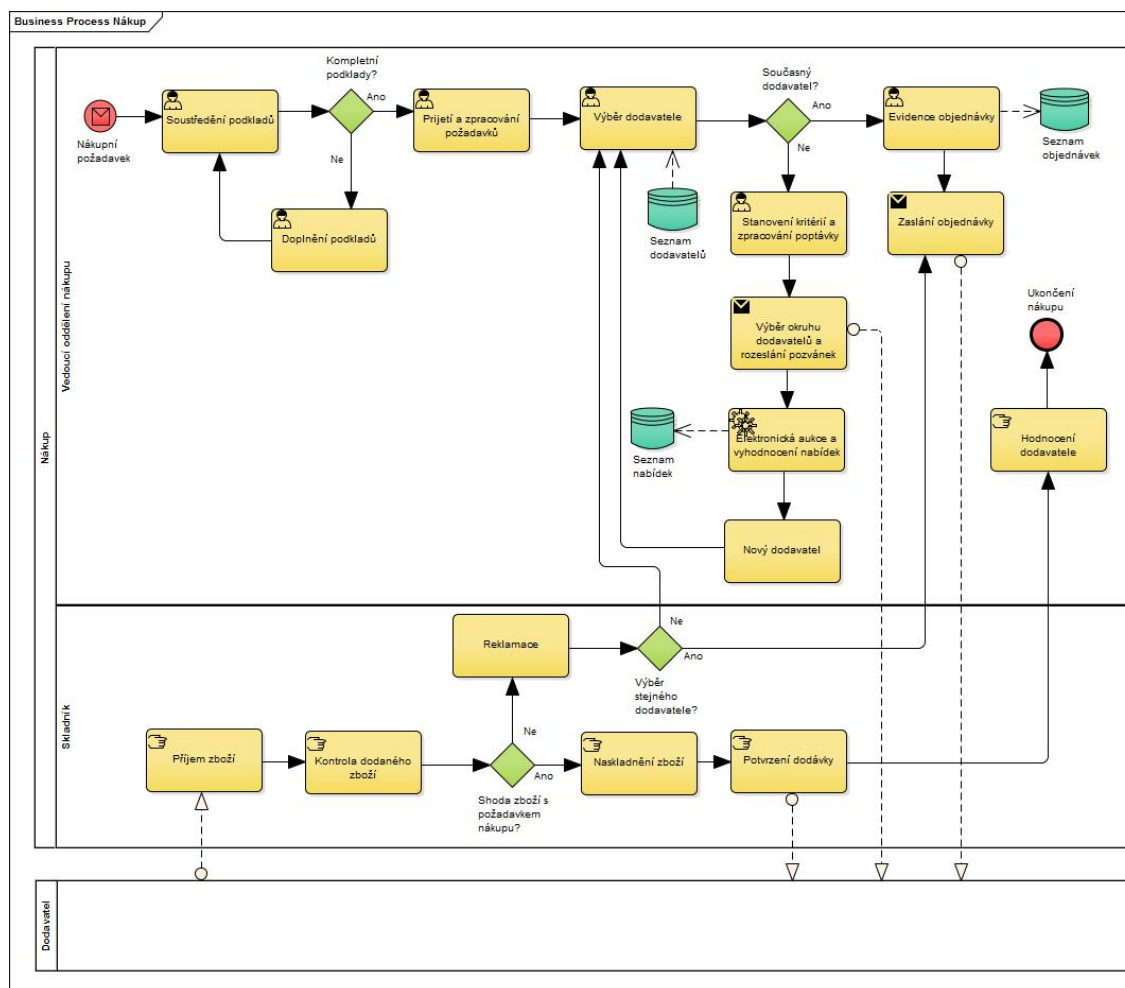
Obr. 8 Průběh procesu Obchod

Na obrázku výše (Obr. 8) lze vidět hlavní proces obchod, který se dělí na prodej z produkce společnosti a nákup surovin pro výrobní činnost.

Cílem obchodního procesu je zabezpečit dostatečný objem zákazníků, kteří budou odebírat produkty společnosti, a také je nutné zajistit jejich spokojenost pro případnou další spolupráci.

Cílem nákupního procesu je obstarat, aby všechny potřebné suroviny či materiály byly nachystány pro výrobní činnost. Nákup zajišťuje vedoucí obchodního oddělení dle požadavků z výroby, kde je sledován stav surovin a materiálu na skladě. Samotný subproces nákup je podrobněji popsán níže.

Subproces Nákup



Obr. 9 Průběh subprocesu Nákup

Pod základní funkci podpůrného procesu nákup si lze představit zabezpečení potřeby, která vzniká z aktivit výrobní činnosti. Výroba má za úkol, aby bylo k dispozici ve správný čas, správné množství materiálu a surovin. V případě, že se množství surovin či materiálu blíží hranici minima, obchodní oddělení zajistí jejich nákup.

Začátkem procesu je přijetí nákupního požadavku ze strany výrobního oddělení na základě přehledu stavu surovin a materiálu na skladě. Proces nákupu materiálu a surovin má v kompetenci vedoucí obchodního oddělení, který nejprve shromáždí poklady na nákup. V případě, že některá z informací pro nákup chybí, vyžádá si doplnění podkladů. Jestliže jsou podklady dostatečné, přijímá požadavek, který zpracovává a ukládá do systému. Následujícím krokem je výběr dodavatele. Jestliže se jedná o nákup surovin, který probíhá opakovaně, vedoucí vybírá dodavatele z dříve nasmlouvaných dodavatelů s ohledem na jejich hodnocení. V případě nových dodavatelů se stanovují kritéria výběru. Poté se vybírá okruh dodavatelů,

keré chce oslovit s poptávkou, a jsou rozeslány pozvánky do elektronické aukce z aukčního systému společnosti, který je výstupem této práce. Dalším krokem je samotná elektronická aukce, ve které soutěží dodavatelé o nejlepší ceny poptávaných surovin. Po skončení elektronické aukce se vyhodnocují nabídky dodavatelů a je vybrán nový dodavatel surovin či materiálu.

Proces nákupu zajišťuje také hodnocení dodavatelů, v rámci kterého se ukládají informace o spokojenosti společnosti s kvalitou odpovídající požadavkům a informace o rychlosti dodávky. Tento přehled dodavatelů a jejich hodnocení má na starosti opět vedoucí obchodního oddělení.

4.3 Problémová doména a současný stav

Webová aplikace se zabývá problematikou nákupu pomocí elektronické aukce uvnitř obchodně-výrobní společnosti a slouží jako podpůrný nástroj v procesu nákupu.

Výsledná webová aplikace tedy bude sloužit k vytvoření elektronických aukcí, ve kterých se budou uvádět nakupované suroviny a materiály. Pomocí elektronické aukce bude obchodní oddělení provádět sběr cenových nabídek dodavatelů a poté samotný výběr dodavatele, kterého zvolí k dodání poptávaných položek.

Při nákupu surovin a materiálu se definují pravidla, podle kterých se vybírá dodavatel. Nejčastějším kritériem výběru je cena položky. Společnost používá systém pro řízení ceníků dodavatelů, který se pravidelně synchronizuje v případě změny cen dodavatelů. Také používá klasickou formu sbírání cenových nabídek, a to pomocí elektronické komunikace a telefonických hovorů.

Po získání cenových nabídek dodavatelů společnost předala nejvýhodnější nabídku poskytovateli, který společnosti zajišťoval outsourcing e-aukcí. Poskytovatel provedl ve svém systému elektronickou aukci s dodavatelí, které jim poskytla společnost, a zároveň oslovil další dodavatele. V případě úspory, však došlo k výpočtu odměny poskytovateli e-aukce. Společnost se proto rozhodla, že odměny za úspory v e-aukci ušetří, pořídí si vlastní systém pro elektronické aukce a tento krok výběrového řízení zajistí vlastními silami i přesto, že nemá všechny zkušenosti a potřebné know-how při složitých a méně častých výběrových řízeních.

Součástí optimalizace nákupního procesu, který ve společnosti probíhá za pomoci elektronické aukce je i centralizace procesu. Cílem centralizace je potřebné snížení nákladů a transparentnost nákupu. Probíhá úplná centralizace nákupu do jednoho oddělení, a to do oddělení obchodu. Jehož strategickým úkolem je zabezpečit, nakoupením surovin a materiálu, zásobování pro společnost.

4.4 Specifikace systémových požadavků na webovou aplikaci

4.4.1 Sběr požadavků

Hlavním požadavkem společnosti a cílem této práce je vytvoření systému pro tvorbu elektronických aukcí. Nově vytvořená webová aplikace bude sloužit jako součást procesu nákupu v obchodně-výrobní společnosti.

Funkční požadavky

Funkčními požadavky je stanovené, co by měl aukční systém vykonávat podle potřeb uživatelů. Hlavním funkčním požadavkem je určené, aby webová aplikace umožňovala tvorbu a správu elektronických aukcí v oblasti nákupu i prodeje obchodního oddělení společnosti. Mezi další požadavky, které aplikace musí splňovat, patří funkce:

- poskytnout zobrazení veřejných informací,
- umožnit registraci a poté přihlášení uživatelů s validací vkládaných dat,
- umožnit zobrazení profilu uživatele, správu těchto údajů včetně možnosti změny hesla,
- umožnit správu kategorií elektronické aukce, jejich přidání, úpravu, odstranění a řazení kategorií,
- umožnit správu měrných jednotek elektronické aukce, jejich přidání, úpravu, odstranění a řazení těchto jednotek,
- poskytovat souhrnný přehled všech registrovaných uživatelů s možností přidání či odebrání rolí jednotlivým uživatelům,
- zaznamenávat práci všech uživatelů s možností zobrazení historie práce vybraných uživatelů,
- umožnit tvorbu a správu elektronických aukcí s možností jejich zveřejnění uživatelům přidáných do elektronické aukce,
- umožnit přidání, upravení a odebrání položek elektronické aukce,
- přidat či odebrat účastníka elektronické aukce,
- zaslat e-mail (pozvánku o elektronické aukci) vybraným účastníkům,
- zobrazit cenové nabídky účastníků s informací o počáteční ceně, nejlepší ceně a úspoře jednotlivých položek,
- umožnit zasílání chatových zpráv v průběhu elektronické aukce,
- poskytnout přehled informací o vytvořených elektronických aukcích, například datum vytvoření aukce, její začátek a konec,
- zobrazovat zbývající čas do vypršení konce elektronické aukce v jejím průběhu,
- umožnit vkládání cenových nabídek,

- umožnit omezený přístup k vypsáním elektronickým aukcím pro vybrané uživatele,
- poskytnout souhrnný přehled informací vybrané elektronické aukce s možností zobrazení informací v tabulkovém procesoru.

Nefunkční požadavky

Nefunkční požadavky jsou omezující podmínky na aukční systém. Tyto požadavky jsou doplňujícími kritérii funkčních požadavků.

Aukční systém je navržen a vytvořen jako webová aplikace přístupná návštěvníkům na Internetu s ohledem na použitelnost. Mezi použitelnost je řazena jednoduchost a intuitivnost aplikace, na tyto požadavky klade důraz zadavatel aplikace. Přestože bude aplikace vystavena na Internetu, neočekává se její vysoká návštěvnost, jako například u webových stránek prezentující společnost. Použití je pouze pro obchodní oddělení společnosti, jako pomocník při nákupu a prodeji zboží.

Důležitá podmínka je stanovena na výkonnost aplikace při zpracování požadavků. Zde je kladen důraz na rychlost aktualizace stránek, kde se zobrazují nejlepší ceny položek v elektronické aukci a chatové zprávy. U těchto stránek se využívá technologie AJAX, která pracuje s částmi stránek, aniž by musela načítat znovu celou stránku, ale pouze její část.

Další podmínkou je škálovatelnost systému, ta musí podporovat rozšiřitelnost aukčního systému na základě potřeb změn či úprav.

Webová aplikace aukčního systému je vytvořena pomocí technologie ASP.NET 2.0, která je součástí .NET Frameworku určeným pro tvorbu webových aplikací a programována jazykem C#. Jako softwarový webový server je použit Internet Information Services (IIS) vytvořený společností Microsoft a databázový server od téže firmy s označením Microsoft SQL Server 2008 R2.

Identifikace uživatelů

Ve webové aplikaci můžeme identifikovat pět různých uživatelů. Návštěvníci, kteří nebudou v aplikaci přihlášení, budou mít přístup pouze k veřejným informacím a dále možnost přihlášení či registraci do systému.

Z nově registrovaných návštěvníků se automaticky stávají dodavatelé. Tito uživatelé mají možnost zobrazit si elektronické aukce, ve kterých jsou přiřazeni. Do těchto aukcí mají možnost vkládat cenovou nabídku položkám v případě, že aukce stále běží, či posílat chatové zprávy vyhlášovatelům v případě dotazů.

Elektronické aukce budou nejčastěji vytvářet vyhlášovatelé. Tito uživatelé jsou registrováni do systému a jejich role jim umožňuje tvorbu elektronické aukce. Při její tvorbě zadávají základní informace aukce, dále položky, které chtějí soutěžit, poté přidávají dodavatele, které chtějí pozvat, a nakonec zasílají pozvánky do aukce. V průběhu mají možnost zasílat doplňující informace dodavatelům pomocí chatové zprávy nebo e-mailové zprávy. Na konci elektronické aukce umožňuje systém uložit tabulkový soubor s veškerými podklady a výsledky.

Dalšími uživateli jsou pozorovatelé elektronické aukce. Tito uživatelé jsou nejčastěji ze strany vedení či jejich zástupců obchodního oddělení. Ti mají zájem prohlížet elektronické aukce bez možnosti zásahu, ale s možností stažení kompletního reportu.

Posledními uživateli jsou administrátoři. Tito uživatelé mají možnost správy měrných jednotek, kategorií, úpravy rolí všech registrovaných uživatelů a v neposlední řadě možnost vytvořit elektronické aukce.

Požadavky na personalizaci

Díky identifikování uživatelů a zjištění, že v aplikaci bude pracovat více skupin uživatelů, lze říci, že webová aplikace bude personalizována. Z těchto důvodů se jeví jako nezbytné přiřazovat uživatele do rolí, kde každá role bude mít určená přístupová práva do webové aplikace a díky tomu budou mít uživatelé přístup jen do určité oblasti aplikace. Část aplikace a její publikované informace budou veřejně k dispozici bez nutnosti registrace a přihlášení do aplikace. Větší oblast bude přístupná jen uživatelům přihlášeným do webové aplikace, kteří díky oprávněním budou mít přístup jen do určitých částí.

Požadavky na data

Část požadavky na data navazuje na předchozí části. A to na části zjištění požadavků na aplikaci, která musí mít požadované funkce a identifikace uživatelů, kteří budou mít požadovaný přístup do aplikace. V této fázi je nutné si ujasnit, která data má aplikace ukládat a jak mají být prezentována. Podrobněji se k datům přistupuje při návrhu datové struktury.

První oblast dat se týká uživatelů webové aplikace. Kde záznamy představují registrované uživatele s jejich podrobnými údaji, včetně údajů společnosti. Do této oblasti dat spadá i správa jejich účtů.

Do druhé oblasti se řadí samotná elektronická aukce. Zde se ukládají její základní informace, které je nutné vyplnit při realizaci aukce. Dále položky elektronické aukce, které představují nákup či prodej surovin a materiálu. Tato oblast zahrnuje i dodavatele přiřazené do aukce včetně jejich cenových nabídek. Chatové zprávy představují další část dat spojené s elektronickou aukcí. Je nutné uvést, že data z této oblasti se dají uložit v tabulkovém souboru a jsou připraveny k využití dalšími aplikacemi. Z těchto dat se evidují nabídky dodavatelů a další informace, které slouží k vnitřnímu hodnocení dodavatelů ve společnosti.

Další oblastí dat bude správa kategorií, do kterých spadají elektronické aukce. Dále správa jednotek položek a správa rolí uživatelů. Poslední záznamy budou informace o práci uživatelů, jejich ukládání pohybu ve webové aplikaci.

4.4.2 Analýza požadavků

Na základě předchozí kapitoly, sběr požadavků, lze provést analýzu požadavků. Analýza požadavků se bude rozdělovat na tři části. A to rozdělení skupin uživatelů, vytvoření diagramů případů užití a scénářů případů užití.

Skupiny uživatelů

Dle požadavků zadavatele na webovou aplikaci bude moci s aplikací pracovat pět různých skupin uživatelů dle jednotlivých rolí. Mezi základní skupinu uživatelů jsou řazeni uživatelé neregistrovaní. Tito návštěvníci stránek si mohou zobrazit pouze veřejné informace, případně se zaregistrovat do systému. Dalšími uživateli jsou již registrovaní uživatelé a uživatelé přihlášení v aplikaci. Ty lze rozdělit dle rolí na vyhlášovatele elektronické aukce, pozorovatele, dodavatele a administrátora webové aplikace.

Dodavatelem se uživatel stane ihned po zaregistrování do aukčního systému. Uživatelů s touto rolí bude v systému většina. Je to uživatel, který může vkládat cenové nabídky položkám poté, co je přiřazen vyhlášovatelem do elektronické aukce. Cenové nabídky má možnost vkládat pouze v době trvání elektronické aukce, po jejím ukončení jsou položky dodavatelům skryté. Mezi další dostupné funkce lze řadit možnost odesílání chatových zpráv vyhlášovateli aukce a jejich přijímání či úpravy profilu včetně hesla.

Uživatelé s rolí vyhlášovatel elektronické aukce jsou osoby z řad obchodního oddělení. Tito uživatele jsou tvůrci elektronické aukce. Při tvorbě elektronické aukce vyplňují základní informace, jako je název, popis, datum začátku a konce, kategorii výběrového řízení. Dále přidávají položky, dodavatele, které chtějí oslovit s možností odeslat jim pozvánku do elektronické aukce. V průběhu elektronické aukce mají možnost sledovat nabídky dodavatelů a komunikovat s nimi pomocí chatových zpráv. Po konci elektronické aukce si mohou stáhnout kompletní informace o celém průběhu aukce v podobě reportu v tabulkovém procesoru. Uživatelé této skupiny mají opět možnost zobrazit si svůj profil s možností úpravy profilu či hesla.

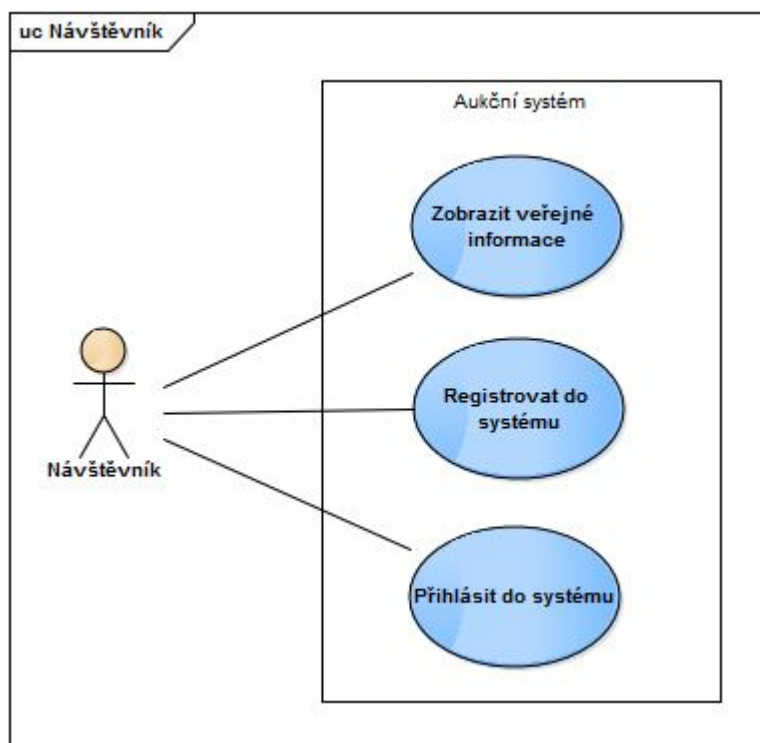
Uživatelé s rolí pozorovatel elektronické aukce jsou opět osoby z řad obchodního oddělení. Tito uživatelé nemají možnost vytvořit elektronickou aukci, ale jejich potřebou je prohlížet vytvořené aukce. Do nich nemohou přímo zasahovat, jako vyhlášovatel, ale pouze prohlížet a případně uložit výsledný report. I oni mají možnost zobrazit svůj profil s možností jeho úpravy či změny hesla.

Poslední požadovanou skupinou jsou administrátoři aplikace. Tuto roli bude využívat zejména vedoucí obchodního oddělení. Uživatel s touto rolí může vytvářet vlastní elektronické aukce, stejně jako vyhlášovatel. Administrátor aplikace má navíc možnost spravovat kategorie elektronických aukcí. Další možností je správa měrných jednotek položek a možnost provádět změny v oprávnění uživatelů. Poslední funkcí je zobrazení historie práce vybraného uživatele.

Diagramy případů užití

V předchozí části jsou definovány skupiny uživatelů, které budou mít k výsledné aplikaci přístup a možnost s ní pracovat. Stanovením seznamu funkčních požadavků jsou určeny základní funkce webové aplikace. Díky těmto získaným informacím lze modelovat případy užití (Use Case) definovaných jazykem UML (Unified Modeling Language). Případy užití jsou tedy specifikace posloupnosti činností, které jsou vykonávány aukčním systémem a vedou k určitému výsledku. Tyto výsledky jsou dů-

ležitě pro účastníka případů užití. Výsledným grafickým návrhem si uživatelé systému ujasní, jaké možnosti nabídne každé roli uživatelů navržený systém.

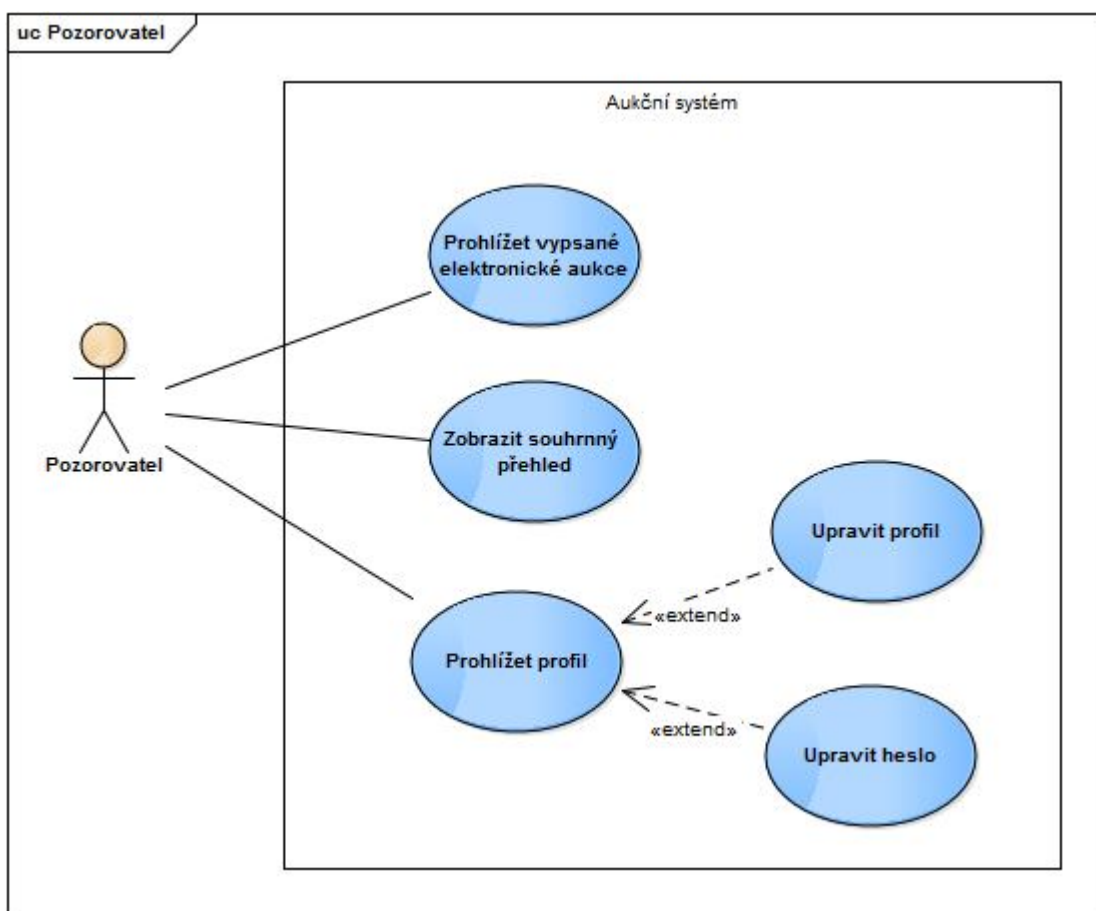


Obr. 10 Diagram případů užití pro návštěvníka

Základní případ užití, který může nejčastěji vzniknout, je případ, kdy webovou aplikaci zobrazí návštěvník stránek. Tento diagram případů užití lze vidět na obrázku výše (Obr. 10). Návštěvník stránek může zobrazit veřejné informace nebo se registrovat do aukčního portálu. V případě, že je již zaregistrován má možnost přihlásit se do systému a pracovat v něm.

Druhým popsáním případem užití je na obrázku (Obr. 11) Use Case diagram pro pozorovatele. Tato role je určena pracovníkům obchodního oddělení. Jejich činností je možnost zobrazení detailů všech vytvořených elektronických aukcí s oprávněním si veškeré informace zobrazit a uložit ve výsledném reportu.

Další namodelované případy užití je možné si prohlédnout v příloze C (Diagramy případů užití) této práce.



Obr. 11 Diagram případů užití pro pozorovatele

Scénář případů užití

Specifikace případů užití je navázání na předchozí kapitolu s názvem Diagramy případů užití. Prostřednictvím diagramů případů užití lze zobrazit vztahy mezi uživateli a funkcemi systému. Podrobný popis chování, tedy k čemu konkrétní případ užití slouží, určuje specifikace případů užití.

Základním scénářem případu užití Návštěvník je registrace do webové aplikace. Tento krok musí být proveden jakýmkoliv uživatelem, pokud chce s aukčním systémem pracovat a dosud v něm není zaregistrován. Specifikace případu užití najdeme v tabulce níže (Tab. 1).

Tab. 1 Scénář případu užití: Registrace uživatele

Případ užití: Registrace uživatele
ID: 1
Stručný popis: Neregistrovaný návštěvník stránek provádí registraci do aukčního systému.
Hlavní aktéři: Neregistrovaný návštěvník stránek.
Vedlejší aktéři: Žádní.
Vstupní podmínky: Žádné.
Hlavní scénář: <ol style="list-style-type: none"> 1. Případ užití začíná, když neregistrovaný návštěvník zvolí příkaz registrace. 2. Systém zobrazí stránku s registračním formulářem. 3. Návštěvník vkládá požadované registrační údaje do formuláře. <ol style="list-style-type: none"> 3.1. Systém ověřuje zadávané údaje návštěvníkem. 3.2. V případě zjištění neplatných či neúplných údajů systém požádá návštěvníka k opravě nebo doplnění. 4. Návštěvník odesílá vyplněné údaje do systému. 5. Systém vytvoří nový účet uživatele a přihlásí ho do systému.
Výstupní podmínky: <ol style="list-style-type: none"> 1. Návštěvník je úspěšně zaregistrován a byl mu vytvořen nový účet.
Alternativní scénáře: Žádné.

Případ užití Zobrazit vytvořené elektronické aukce, který je podrobně popsán v tabulce (Tab. 2), slouží pro získání přehledu vytvořených elektronických aukcí uživatelem. Tento případ užití obsahuje místo rozšíření Upravit elektronickou aukci.

Tab. 2 Scénář případu užití: Zobrazit vytvořené elektronické aukce

Případ užití: Zobrazit vytvořené elektronické aukce
ID: 2
Stručný popis: Uživatel si může zobrazit přehled svých vytvořených elektronických aukcí.
Hlavní aktéři: Uživatel.
Vedlejší aktéři: Žádní.
Vstupní podmínky: Pro zobrazení vytvořených elektronických aukcí je nutné být přihlášen do aukčního systému a mít přiřazenu roli Vyhlašovatel.
Hlavní scénář: <ol style="list-style-type: none"> 1. Případ užití je spuštěn příkazem Vypsání aukce. 2. Systém uživateli zobrazí stránku se seznamem vypsání elektronických aukcí a jejich základními informacemi. 3. Uživatel si může u vybrané elektronické aukce zobrazit její detailní informace společně s přidáním dodavatelů, položkami aukce a nabídkami dodavatelů. Místo rozšíření: Upravit elektronickou aukci.
Výstupní podmínky: Žádné.
Alternativní scénáře: Žádné.

Rozšiřující případ užití Upravit elektronickou aukci, který je popsán v následující tabulce, slouží pro změnu dat vytvořené aukce. Je využíván v okamžiku, kdy vyhlášovatel špatně zadal údaje, či vedoucí obchodního oddělení s takto vytvořenou elektronickou aukcí nesouhlasí a žádá její úpravu.

Tab. 3 Scénář případu užití: Upravit elektronickou aukci

Rozšiřující případ užití: Upravit elektronickou aukci
ID: 3
Stručný popis: Uživatel může u vybrané elektronické aukce provést její úpravy.
Hlavní aktéři: Uživatel.
Vedlejší aktéři: Žádní.
Vstupní podmínky: Pro možnou úpravu je nutný předchozí výběr konkrétní elektronické aukce, která má být změněna.
Hlavní scénář: <ol style="list-style-type: none"> 1. Případ užití je spuštěn uživatelem na příkaz Upravit aukci. 2. Aukční systém uživateli zobrazí webovou stránku s formulářem pro úpravu elektronické aukce. 3. Uživatel provádí požadované úpravy elektronické aukce. <ol style="list-style-type: none"> 3.1. Systém ověřuje zadávané údaje uživatelem. 3.2. V případě zjištění neplatných či neúplných údajů při úpravě elektronické aukce systém požádá uživatele k opravě nebo doplnění. 4. Uživatel odesílá vyplněné údaje do systému. 5. Systém provede úpravy nastavení elektronické aukce.
Výstupní podmínky: Systém úspěšně upravil nastavení elektronické aukce.
Alternativní scénáře: Žádné.

4.5 Datová struktura

Datový model je vytvořen jako konceptuální datový model, který představuje zobecnění oproti struktuře v relační či objektové databázi. Zobecněním získáme nezávislost na konkrétním databázovém systému. Konceptuální datový model je reprezentován jako XML dokument, který slouží k vygenerování datové struktury pro konkrétní systém.

Datový model je vytvořen pomocí nástroje WebRatio a svou strukturou odpovídá požadavkům zadavatele. Navržený výsledný model se nachází v textu níže.

Při vytváření nového modelu nástroj WebRatio automaticky vytvoří tři entity, které jsou pojmenovány User, Group a Module. Tyto entity včetně jejich relací nelze smazat, ale pouze přejmenovat. Pro účely aplikace je použita entita User, entita Group je přejmenována na Roles a entita Module bude v databázové struktuře nevyužita. Mezi další odlišnosti datového modelu patří kardinality, které jsou připsány k entitám v opačném směru, než je obvyklé při tvorbě ER diagramu. Při použití tohoto nástroje a vytváření nové entity se automaticky generuje atribut s názvem

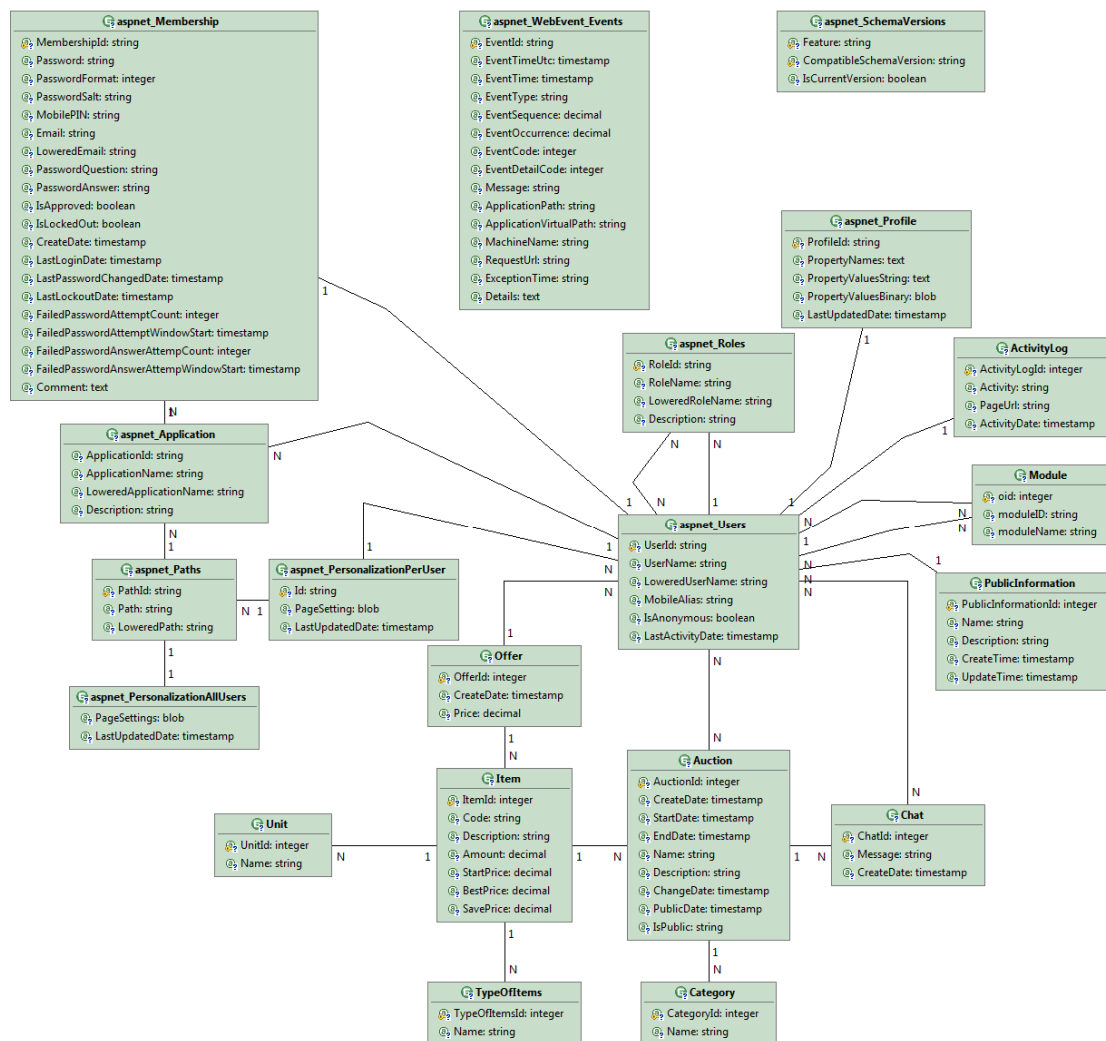
OID, který představuje přirozené číslo. Toto přirozené číslo značí jednoznačně instanci vybrané entity. V modelu nejsou definovány cizí klíče z důvodu nezávislosti na konkrétním systému. Proto nejsou vazební entity v modelu definovány.

Řešená webová aplikace má základní databázovou strukturu vytvořenou pomocí .NET Frameworku nazvanou Microsoft ASP.NET 2.0 Provider Database. Ta je optimálně navržena a automaticky vytvořena při tvorbě aplikace s možností registrace uživatelů a správy jejich účtů. Mezi nejdůležitější entity této databáze patří:

- aspnet_Applications – eviduje informace o aplikaci,
- aspnet_Membership – umožňuje ukládat podrobné informace o uživateli, jako např. datum vytvoření uživatele, heslo, e-mail, datum posledního přihlášení, blokace účtu apod.,
- aspnet_Profile – umožňuje ukládat doplňující údaje o uživateli, v konkrétním případě webové aplikace se jedná o registrační údaje společnosti,
- aspnet_Roles – eviduje uživatelské role, kterými jsou například administrátor, vyhledávací, dodavatel,
- aspnet_Users – entita představuje registrované uživatele webové aplikace.

V diagramu dále rozlišujeme entity týkající se elektronické aukce. Nejdůležitější entitou je Auction ukládající veškeré informace o vytvořených aukcích. Následuje entita AuctionMember, která je navázána na předchozí entitu a ukládá informace o přidávaných účastnících do jednotlivých elektronických aukcí. Entita Category ukládá záznamy o kategoriích, do kterých jednotlivé elektronické aukce patří. Entita Item ukládá informace o položkách a entita Offer eviduje nabídky dodavatelů v rámci položek. Na entitu Item navazuje entita Unit, která ukládá všechny jednotky položek, a entita TypeOfItems, která uchovává dvě instance, a to nákup a prodej. Poslední zbylou entitou týkající se elektronické aukce je Chat, tato entita ukládá veškeré odeslané zprávy uživatelů.

Zbylé entity v datové struktuře jsou ActivityLog a PublicInformation. První slouží k uchování aktivity uživatelů při práci v aplikaci a druhá k ukládání informací, které se budou publikovat uživatelům.

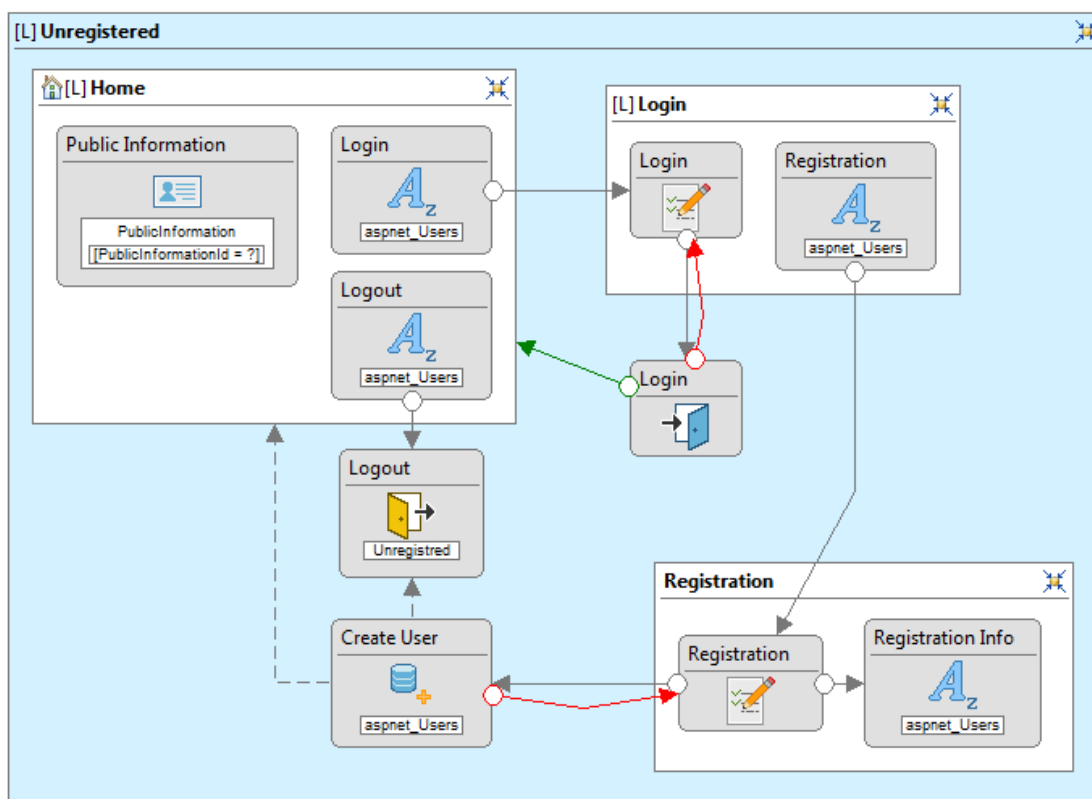


Obr. 12 Datová struktura systému

4.6 Tvorba IFML modelů

Front-end model společně s modelem business logiky tvoří IFML model aplikace. Tento model představuje velmi důležitou část procesu analýzy a návrhu tvorby webové aplikace. IFML model bude definovat sadu webových stránek včetně propojení obsahu pomocí odkazů. V případě jednoho komplexního IFML modelu by zobrazení bylo velmi nepřehledné. Z tohoto důvodu bude IFML model rozdělen na pět částí, podle jednotlivých rolí uživatelů. Grafické zobrazení IFML modelů je součástí přílohy D (IFML modely webové aplikace) této práce.

4.6.1 IFML model pro návštěvníka webové aplikace



Obr. 13 IFML model pro návštěvníka webové aplikace

IFML model pro návštěvníka webové aplikace vychází svoji podstatou z modelu případů užití pro návštěvníka. Tento model webové aplikace je přístupný všem návštěvníkům stránek. Skládá se ze tří stránek. Domovské stránky, která slouží k publikování veřejných informací. Jedná se o stránku, která se automaticky zobrazí při zadání adresy webové aplikace. Dále je to stránka přihlášení, umožňující vstup do neveřejné části, a která kontroluje, zda jsou vyplněny potřebné údaje a jestli je uživatel vyplnil správně. Poslední stránkou je registrace do webové aplikace. Při registraci návštěvník vyplňuje potřebné údaje, které se odesílají ke zpracování. V případě neplatně vyplněných údajů aplikace vyzve uživatele k jejich opravě, v druhém případě, je uživatel zaregistrován do webové aplikace a současně přihlášen.

4.6.2 IFML model pro vyhledávatele elektronické aukce

IFML model pro vyhledávatele elektronické aukce je založen na roli vyhledávatele. Po přihlášení do webové aplikace bude nejčastěji používat stránku vytvořit aukci nebo vypsané aukce. V případě první volby vytvořit novou elektronickou aukci se mu zobrazí základní formulář pro vyplnění dat. Po odeslání dat se vytvoří nová aukce, do které může uživatel přidávat nákupní či prodejní položky dle potřeb. Po

přidání položek následuje funkce přidání a odebrání dodavatelů. To jsou účastníci elektronické aukce, které chceme oslovit, aby nám vložili cenovou nabídku položek. Pro jejich kontaktování slouží formulář pro odeslání e-mailu neboli pozvánek. V průběhu aukce se zobrazuje historie nabídek dodavatelů včetně nejlepších cen položek a ubíhajícího času do konce aukce. Uživatel dále může zaslat dodavatelům chatovou zprávu s doplňujícími informacemi či dotazem. Po skončení elektronické aukce má vyhlášovatel možnost si uložit kompletní informace o průběhu aukce v podobě reportu. V případě volby vypsané aukce si může zvolit vytvořenou aukci k detailnímu prohlížení či úpravě. Dále může vykonávat aktivitu prohlížení svého profilu, včetně úpravy profilu či změny hesla.

4.6.3 IFML model pro pozorovatele elektronické aukce

IFML model pro pozorovatele elektronické aukce je blízký předchozímu modelu pro vyhlášovatele elektronické aukce. Tento model zobrazuje skutečnost, že uživatel s rolí pozorovatel má možnost pouze zobrazovat všechny elektronické aukce s funkcí uložit výsledný report, ale bez možnosti jejich úpravy. Také funkce vytvoření elektronické aukce zde není. Stejně jako předchozí model pro vyhlášovatele, i tento model umožňuje pozorovateli zobrazit svůj profil společně s jeho úpravou či změnou hesla.

4.6.4 IFML model pro dodavatele

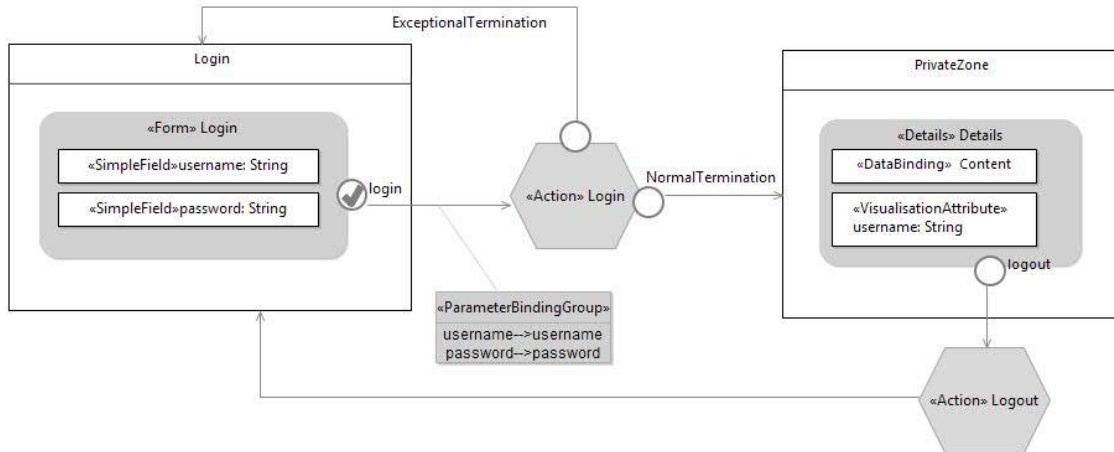
Tento IFML model říká, že uživatel s touto rolí má možnost přistupovat k vytvořeným aukcím v případě, že je do elektronické aukce přizván. V detailním zobrazení vybrané aukce poté vidí jednotlivé položky s možností vkládání cenových nabídek. Tyto nabídky poté může upravovat, ale pouze v průběhu trvání aukce. Současně má k dispozici i funkci odesílání chatových zpráv vyhlášovateli aukce. Zobrazení svého profilu s možností úpravy profilu a změnou hesla je standardní funkce všech registrovaných uživatelů bez rozdílu role.

4.6.5 IFML model pro administrátora webové aplikace

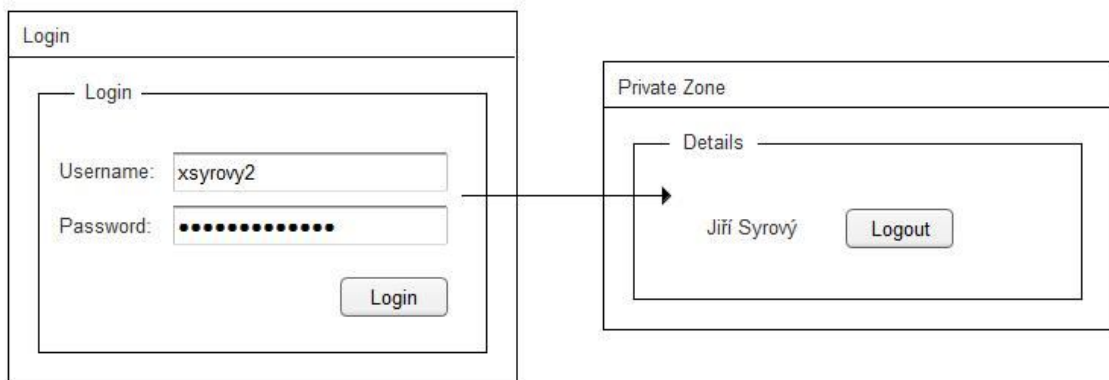
Poslední IFML model je pro administrátora webové aplikace. Jeho hlavní funkcí není možnost založení aukce, i když, že tuto možnost má. Tento model definuje, že hlavní oblastí zájmů je správa kategorií, do kterých se elektronické aukce řadí a dále správa jednotek položek. Administrátorovi je dále přidána možnost spravovat role všem registrovaným uživatelům aplikace. Poslední funkcí, kterou lze vidět na modelu, je možnost zobrazení historie práce vybraného uživatele a správa vlastního profilu.

Zmíněné modely jsou vytvářeny v komerčním vývojovém prostředí WebRatio. Tomuto prostředí v prosinci roku 2014 přibyl konkurent v podobě open source vývojového prostředí Eclipse. S rozšířením pomocí pluginu je možné vytvářet i v tomto nástroji IFML modely. Mezi nástroji lze najít určité odlišnosti. WebRatio podporuje více komponent, které odlišuje graficky. Pomocí Eclipse jde zase komponentám přidat atributy a tím komponenty zpřesnit. WebRatio tuto možnost také

umožňuje, ale vizuálně se v komponentě atributy nezobrazují. V modelech jsou vidět i jiné odlišnosti, příkladem lze uvést předání parametrů či rozdílné značení modelů pro mobilní aplikace. Ukázky modelů pomocí nástroje Eclipse jsou na obrázcích níže.

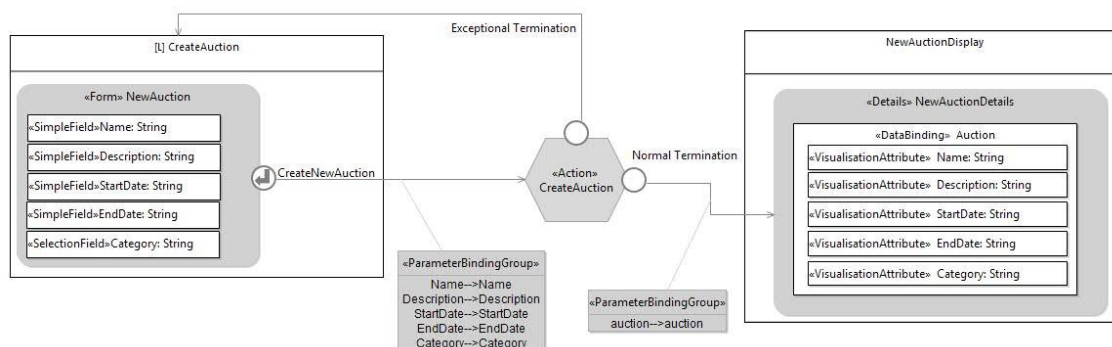


Obr. 14 IFML model přihlášení uživatele

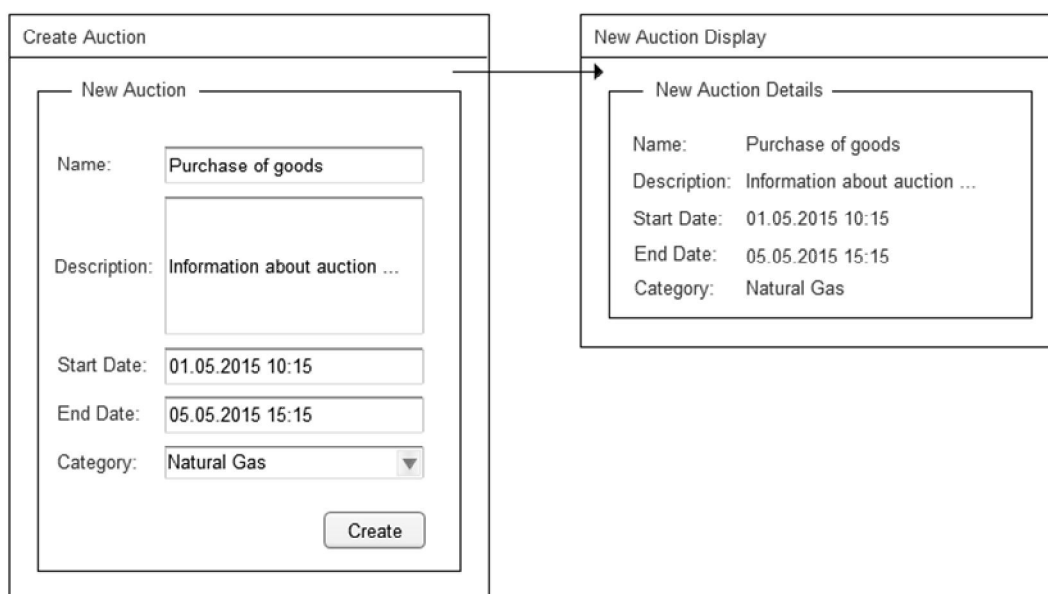


Obr. 15 Drátěný model přihlášení uživatele

Na obrázcích (Obr. 14, Obr. 15) lze vidět ukázkou IFML modelu pro přihlášení uživatele do webové aplikace. V případě, že uživatel zadá správné přihlašovací údaje, aplikace ho přesměruje na stránky, které jsou k dispozici pouze po přihlášení. V případě nesprávných údajů ho vyzve k opravě údajů.



Obr. 16 IFML model vytvoření elektronické aukce



Obr. 17 Drátěný model vytvoření elektronické aukce

Na snímcích (Obr. 16, Obr. 17) lze vidět vytvoření elektronické aukce. Po zadání povinných údajů a odeslání formuláře se dostaneme na stránku s detailním zobrazením aukce, která slouží pro další práci.

4.7 Prezentční návrh

Na prezentační návrh webové aplikace nejsou kladeny velké požadavky. Společnost má jediný požadavek, aby aplikace byla pojata v barvách firemního stylu bez zbytečných grafických prvků. Je to z důvodu přehlednosti a jednoduchosti aplikace. Předpokládá se, že nejčastěji budou návštěvníci aplikace ze stran dodavatelů. Tito uživatelé dle společnosti nejsou často počítačově gramotní, proto funkčnost i vzhled musí být co možno nejjednodušší. Cílem aplikace také není zvýšení návštěvnosti, touto problematikou se věnují webové stránky prezentující společnost.

Registrujte se pokud nemáte účet.'" data-bbox="163 110 873 406"/>

Obr. 18 Přihlášení uživatele do webové aplikace

Na obrázku výše (Obr. 18) lze vidět přihlašovací obrazovku uživatele do webové aplikace. Další náhledy aplikace je možné najít v příloze A (Náhled jednotlivých částí webové aplikace) a na přiloženém CD.

4.8 Implementace webové aplikace

Implementace webové aplikace je založena na předchozím vymezení systémových požadavků. Funkční požadavky jsou stanovením toho, co by měl aukční systém vykonávat dle potřeb uživatelů. Při implementaci bereme v úvahu i vytvořené modely a veškeré specifikace. Nefunkčními požadavky jsou určeny technologie, programovací jazyk a databázový server, které se mají při tvorbě aplikace zvolit. Průběh programování webové aplikace není možné ani nutné do detailů popisovat.

Prvním krokem implementace webové aplikace je vytvoření návrhu datové struktury v nástroji WebRatio. Protože v tomto nástroji se nepodařilo vytvořit funkční skript pro Microsoft SQL Server, byla použita aplikace Microsoft SQL Server Management Studio, která je administračním nástrojem pro návrh struktury databáze a umožňuje pracovat s uloženými daty. Oproti návrhu je implementace odlišná v kardinalitách, které se v navržené struktuře připisují v opačném směru, než je obvyklé. Dále bylo nutné definovat cizí klíče a vazební entity, které nebyly z důvodu nedefinovaných cizích klíčů modelovány.

Po vytvoření datové struktury následovalo samotné programování webové aplikace ve vývojovém prostředí Microsoft Visual Web Developer, kde byl zvolen projekt ASP.NET Web Site a vybrán jazyk C#. Při vývoji byly dále využity IFML modely, které znázorňují webové stránky včetně propojení obsahu pomocí odkazů.

Při vývoji funkční logiky webové aplikace je kód uložen do *code behind* souborů stránek s příponou *.aspx.cs*. Uložené stránky do dvou souborů mají výhodu v tom, že první soubor obsahuje pouze HTML kód stránky, do kterého jsou vloženy značky serverových komponent. V druhém souboru jsou například ošetřeny události komponent, nastavení vlastností či práce s daty. Pro samotnou práci s daty byl použit integrovaný jazyk pro dotazování zvaný LINQ.

Výsledné řešení webové aplikace je označeno *AS2015*. Strukturu všech adresářů a souborů aplikace lze prohlížet na přiloženém CD. Zde jsou stručně popsány jednotlivé části:

- **AS2015** – adresář, který obsahuje výsledné řešení webové aplikace. Součástí jsou adresáře a samostatné stránky popsané níže. V adresáři jsou tři soubory, které nedefinují žádnou stránku, ale jsou důležité pro aplikaci. První z nich je soubor *Web.config*, který konfiguruje aplikaci a je založen na formátu XML. V tomto souboru je nastaveno například databázové připojení, autentizace, údaje o uživateli a společnosti, které se mají ukládat či SMTP server. V druhém souboru *Web.sitemap*, který se musí nacházet v kořenovém adresáři aplikace, jsou hierarchicky uspořádány stránky aplikace. U každé stránky se ukládá odkaz, titulek a role. Tímto souborem lze vytvořit menu webové aplikace. Poslední soubor má jméno *Site.master*. Tento soubor slouží k začleňování stránek do společného vzhledu webové aplikace.
- **Account** – adresář obsahující stránky pro registraci, přihlášení, změnu hesla, prohlížení a úpravu profilu.
- **Admin** – obsahuje soubory pro správu kategorií a jednotek, dále pro správu rolí uživatelů a historii jejich práce.
- **App_Code** – nachází se zde třídy obsahující kód aplikace. Dále se zde nachází soubory s příponou *.edmx* a *.Designer.cs*. Tyto soubory převádí vytvořenou datovou strukturu pro použití integrovaného jazyka umožňujícího dotazování, který je nazvaný LINQ (Language Integrated Query).
- **App_Data** – zde webová aplikace ukládá soubor s databází aplikace.
- **Auctions** – adresář pro dodavatele obsahující stránku se seznamem aukcí, detail jednotlivé elektronické aukce a stránku pro vkládání nabídek.
- **Bin** – adresář pro umístění knihoven použitých ve webové aplikaci. Ve výsledné aplikaci je použita knihovna *AjaxControlToolkit*.
- **MyAuction** – největší část implementované webové aplikace. V tomto adresáři se nachází soubory pro vytvoření nové aukce a její případné editace. Dále soubory pro přidání, odebrání a editaci položek, přidání i odebrání dodavatelů. Nesmíme zapomenout i na soubory řešící seznam vytvořených aukcí.
- **Scripts** – adresář pro přiložené soubory s JavaScriptem.
- **Styles** – obsahuje soubor pro uložení výsledného vzhledu webové aplikace.

Webová aplikace pro živou ukázkou výstupu této práce je dostupná na adrese: <http://auction.aspone.cz/>. Pro finální nasazení webové aplikace je určen server společnosti s přístupným databázovým serverem.

4.9 Testování

Notace IFML nemá přesně definovanou fázi testování, neurčuje, jak by mělo testování probíhat. Samotné testování probíhá vždy při vytvoření nového prototypu a probíhá průběžně při vývoji webové aplikace.

Testují se zejména funkční a nefunkční požadavky. Kontrola se provádí v součinnosti se společností, při které se ověřuje, zda stanovené požadavky jsou správně aplikovány. Zároveň probíhá testování aplikace rozdělené do několika oblastí. První oblast se zabývá funkčním testováním (kontrola odkazů, kontrola vyplňování formulářů a komunikace s databází). Další oblastí je testování použitelnosti (testování navigace a interakce uživatele s prostředím aplikace), při každém prototypu se testuje i výkon aplikace. Zde se primárně kontroluje pole VIEWSTATE. Toto pole obsahuje informace změněných hodnot vlastností a v případě složitých stránek může zabírat mnoho místa. Tyto informace se poté odesílají na server a zpět, tímto se komunikace mezi klientem a serverem může zpomalit. Tento stav je nepříjemný, musí se sledovat a v případě navýšení velikosti pole se komponenty upravují.

4.10 Nasazení, údržba a rozvoj aplikace

Pro nasazení výsledné webové aplikace společnost vyhradila server s operačním systémem Windows Server 2008 R2. Databáze je zprovozněna na Microsoft SQL Serveru 2008 R2.

Konečná webová aplikace po domluvě se společností je předána na IT oddělení, které zajistí její pravidelnou údržbu a bude se starat o chod webové aplikace. Mezi tuto údržbu spadá i hardwarové vybavení a zajištění připojení serveru k Internetu.

Webová aplikace řeší problematiku nákupu zboží pomocí elektronické aukce. Splňuje definované funkční požadavky a je předána společnosti k využití. Její možný rozvoj záleží na vzniklých požadavcích a připomínkách, jak ze strany vyhlásovaatele elektronické aukce, tak i ze strany dodavatelů.

5 Diskuze

Úvodem byla v této práci zmíněna teoretická východiska problematiky týkající se webových aplikací. Díky dostupnosti, kvalitě a rychlosti Internetu se softwarové produkty, které byly dříve pouze na klientském počítači, přenáší do prostředí Internetu. Kvalitních metodologií pro tvorbu softwaru existuje velké množství, mezi nimi lze najít univerzální jazyk UML. Tento jazyk ale není úplně vhodný pro tvorbu webových aplikací, protože mu chybí hypertextového paradigma.

Z tohoto důvodu vznikají modelovací jazyky určené speciálně pro vývoj webových aplikací. V této práci jsou zmíněné přístupy pro modelování webových aplikací a tři z nich podrobněji rozepsány. Mezi popsané metodologie patří OOHDM (Object-Oriented Hypermedia Design Model), WSDM (Web Semantic Design Method) a UWE (UML-based Web Engineering). Přístup UWE používá při modelování jazyk UML, který rozšiřuje o hypertextové paradigma. Dále byl zmíněn přístup k vývoji aplikací pomocí metodologie WebML (Web Modeling Language), která je předchůdcem metodologie IFML (The Interaction Flow Modeling Language).

IFML se stala hlavní metodologií této práce, pomocí které je webová aplikace vyvíjena. IFML je výsledkem výzkumné práce italské univerzity Politecnico di Milano a hlavním tvůrcem Brambillem. Tato metodologie je mladá, konečná verze je přijatá standardizační komisí OMG v březnu 2014. Z tohoto důvodu je i málo publikovaná a rozšířená. Inspiraci převzala z WebML, kterou rozšířila o možnost vývoje aplikací pro mobilní telefony a rozdělila podporu životního cyklu do více modelů.

Oblasti modelování webových aplikací pomocí notace IFML předchází v této práci část, ve které je představena společnost, předmět podnikání a je zobrazena organizační struktura. Na základě rozhovorů s vedením společnosti a získaných informací jsou identifikovány procesy, které jsou rozděleny do hierarchicky uspořádaných úrovní. Mezi hlavní procesy patří Obchod a Výroba, díky nim vzniká přidaná hodnota ve společnosti a podpůrné procesy, jejichž cílem je zajistit fungování hlavních procesů. V této práci je poté popsán proces Obchod a jeho subprocess Nákup, ve kterém se využívá výsledná webová aplikace jako nástroj pro elektronické aukce.

Dříve zajišťoval elektronickou aukci pro společnost poskytovatel outsourcingu. Protože za úsporu v elektronické aukci musela společnost platit poskytovateli provizi, rozhodla se pro systém, ve kterém si bude elektronické aukce vypisovat samostatně. Před samotnou realizací webové aplikace společnost poptávala aukční systém u poskytovatelů, kteří ho nabízejí k pronájmu či odkoupení. Podle rozhovorů byly zásadní problémy ve složitosti systémů, které byly cenově v jiné kategorii, než společnost poptávala, nebo naopak příliš jednoduché, které neumožňovaly v budoucnu případné rozšíření. Na základě těchto skutečností se rozhodla o vlastní řešení systému. Systém je tedy vytvořen přímo pro potřeby společnosti. Jedná se o systém, který nepodléhá zadávání zakázek dle zákona o veřejných zakázkách. Jedná se o webovou aplikaci, kde výběrová řízení jsou realizována formou elektronických aukcí pro podání nabídek.

Samotnému modelování předchází část specifikace systémových požadavků na webovou aplikaci. Tato fáze je definována v životním cyklu metodologie IFML, ale metodologie ani vývojový nástroj WebRatio žádný model neumožňuje vytvořit. Pro tuto fázi je lepší využít jazyk UML, vytvořit diagramy užití a také specifikaci případů užití.

První model, který se vytváří, je doménový model. Při vytváření doménového modelu lze použít jazyk UML. Nástroj WebRatio vytváří konceptuální model, který lze převést do konkrétní relační databáze.

Samotná realizace modelu pomocí WebRatio je poměrně obtížná a to z následujících důvodů. WebRatio automaticky vytvoří tři entity a relace mezi entitami. Entity ani relace však nejdou smazat, entity lze pouze přejmenovat. Další odlišnosti lze spatřit v kardinalitách, které jsou k entitám značeny v opačném směru než při tvorbě ER diagramů. Mezi další odlišnosti lze uvést automaticky generovaný atribut k jednotlivým entitám či nedefinování cizích klíčů z důvodu nezávislosti na konkrétní relační databázi. Z toho důvodu nejsou v datové struktuře vazební tabulky.

Spojením modelů front-end a business logiky do výsledného IFML modelu lze namodelovat potřebné části aplikace i celou webovou aplikaci. Webová aplikace se skládá z jednotlivých kontejnerů a navigací mezi stránkami, kterými lze určit strukturu systému. Přidáním jednotlivých komponent a operací můžeme webové stránky detailněji přiblížit zadavateli i případným spolupracovníkům. Jak podrobné modely budou, záleží na daném systému. V případě velmi složitých aplikací i vytvářených modelů se mohou stát předané modely nepřehlednými a pro případnou úpravu či doplnění neuskutečnitelnými. Tímto by ztratily svoji srozumitelnost, díky které mohou být modely konzultovány napříč společnostmi.

Metodologie IFML navazuje na předchozí metodologii WebML, kterou rozšířila hlavně o možnost vytvářet modely pro mobilní aplikace. Také IFML je implementována v komerčním vývojovém prostředí WebRatio. Tento nástroj poskytuje tvorbu modelů, generování kódu webové aplikace a dokumentace či týmové spolupráce. V prosinci 2014 byl poté vyvinut i zásuvný modul pro open source vývojové prostředí Eclipse. Pomocí tohoto vývojového prostředí chtějí vývojáři metodologii rozšířit oproti předchůdci, metodologii WebML. Porovnáním těchto nástrojů lze konstatovat, že možnosti jsou velmi podobné, ale existují určité odlišnosti. Nástroj WebRatio podporuje více komponent, které jsou graficky odlišeny. Pomocí Eclipse lze přidat komponentám atributy a tím komponenty zpřesnit. WebRatio tuto možnost také umožňuje, ale vizuálně atributy v komponentě nezobrazuje. Mezi další odlišnosti lze uvést značení předání parametrů mezi kontejnery a znázornění modelů pro mobilní aplikace.

Dle získaných poznatků lze napsat, že metodologie IFML se jeví jako vhodný přístup k vývoji webových aplikací. Její rozšíření oproti WebML či ostatním metodologiím je v možnosti vývoje různých typů aplikací, mezi které patří desktopové, webové či mobilní. Kompletní proces vývoje aplikace patří mezi další kladné stránky. Jelikož jde o notaci mladou, lze očekávat případné rozšíření mezi další

vývojová prostředí. Její využití závisí na rozsahu projektu, nákladech či časové náročnosti.

Výsledná webová aplikace byla nahrána na server společnosti, která vyhradila pro tento systém Windows Server 2008 R2 s databází SQL Server 2008 R2. Výhodou nasazení společnost uvádí, že aplikace funguje na stávajícím hardwarovém a softwarovém vybavení, proto nebyly nutné investice do těchto prostředků. Před samotným nahráním na server byl systém nejprve podrobně otestován v součinnosti se všemi pracovníky obchodního oddělení. Vedoucí obchodního oddělení vytvořil elektronické aukce na nákup části potrubí a ostatní členové oddělení byli testovacími dodavateli. Aplikace byla kladně hodnocena a členové oddělení ocenili i možnost vyzkoušení si podání nabídky ze strany dodavatelů. Toto testování bylo i v rámci zaškolení zaměstnanců, kteří budou systém dále využívat.

Mezi již naplánované rozšíření aplikace se řadí nápověda systému a jazykové mutace, díky kterým chce společnost rozšířit oblast možných dodavatelů surovin a materiálů. Další funkcí, kterou by společnost ocenila, jsou globální statistiky ze všech elektronických aukcí s podrobnými výsledky a úsporami. Další rozvoj záleží na vzniklých požadavcích a připomínkách, jak ze strany provozovatele systému, tak i ze strany dodavatelů.

6 Závěr

Hlavním cílem této práce bylo vyvinout webovou aplikaci řešící problémovou doménu pomocí modelovacího jazyka IFML a CASE nástroje WebRatio. Systém podporující proces nákupu surovin a materiálu byl rozšířen o možnost elektronické aukce. Pomocí elektronické aukce bude obchodní oddělení provádět sběr cenových nabídek dodavatelů a poté samotný výběr dodavatele, kterého zvolí k dodání požadovaných položek.

Práce byla rozdělena do dvou částí, a to na část teoretickou a část vlastní, praktickou. V teoretické části práce bylo nutné se zaměřit na problematiku webových aplikací a technologií. Technologií pro vývoj webových aplikací existuje velké množství. Byl vybrán zástupce ASP.NET, jehož technologie byly popsány, a pro samotný vývoj vybrán framework Web Forms a databázový nástroj SQL Server od firmy Microsoft.

Po získání povědomí o této problematice bylo přistoupeno k nastudování metodologií pro vývoj webových aplikací. Nejprve bylo nutné se zaměřit na metodologii WebML, která je předchůdce metodologie IFML. Modelovací jazyk IFML byl poté nastudován a podrobně popsán. Byl rozebrán proces vývoje webových aplikací pomocí notace IFML. Proces se skládá z následujících částí: specifikace požadavků, datového modelu, front-end modelu, modelu business logiky, prezentačního modelu, implementace, testování a z části nasazení a údržba. Tyto fáze jsou v teoretické části podrobně rozepsány a poté při vývoji samotné aplikace dodrženy.

Metodologie IFML a WebML nepatří mezi jediné možnosti při vývoji webových aplikací. Proto byly zjištěny další přístupy vhodné pro tento vývoj. Vedle IFML a WebML byly nastudovány a v práci uvedeny další přístupy k tvorbě webových aplikací, mezi které patří Object-Oriented Hypermedia Design Method (OOHDM), Web Semantic Design Method (WSDM) a metoda UML-based Web Engineering (UWE).

Poslední teoretická část se zabývala vývojovým nástrojem WebRatio. Tento komerční nástroj se stal prvním prostředím podporující notaci IFML. Dalším možným nástrojem je open source Eclipse se zásuvným modulem, který rozšíří nástroj o návrh IFML.

Teoretická část se stala základem pro vlastní část práce. V této části nejprve byla představena společnost, předmět podnikání a její organizační struktura. Poté byla provedena analýza současných procesů v obchodním oddělení společnosti a popsán proces Nákup. Právě pro tento proces byla navržena a implementována inovace související s nákupem zboží pomocí elektronických aukcí. Postup vlastní práce byl v souladu s jednotlivými fázemi vývoje webové aplikace. V části sběr a analýza požadavků byl využit jazyk UML pro diagramy případů užití, na které se navázalo při tvorbě modelů v dalších fázích. Následně byla vytvořena datová struktura, na kterou navázaly jednotlivé IFML modely. IFML modely se skládají z více modelů, mezi ně patří kompoziční a navigační model a práce s daty. Vytvořené modely poté sloužily při implementaci webové aplikace, k jejíž realizaci byla využita tech-

nologie ASP.NET a databázový nástroj SQL Server. Výsledná webová aplikace byla na závěr testována a nasazena.

Získané informace byly poté diskutovány v části diskuze této práce. Závěrem práce lze říci, že stanovený cíl byl splněn.

7 Literatura

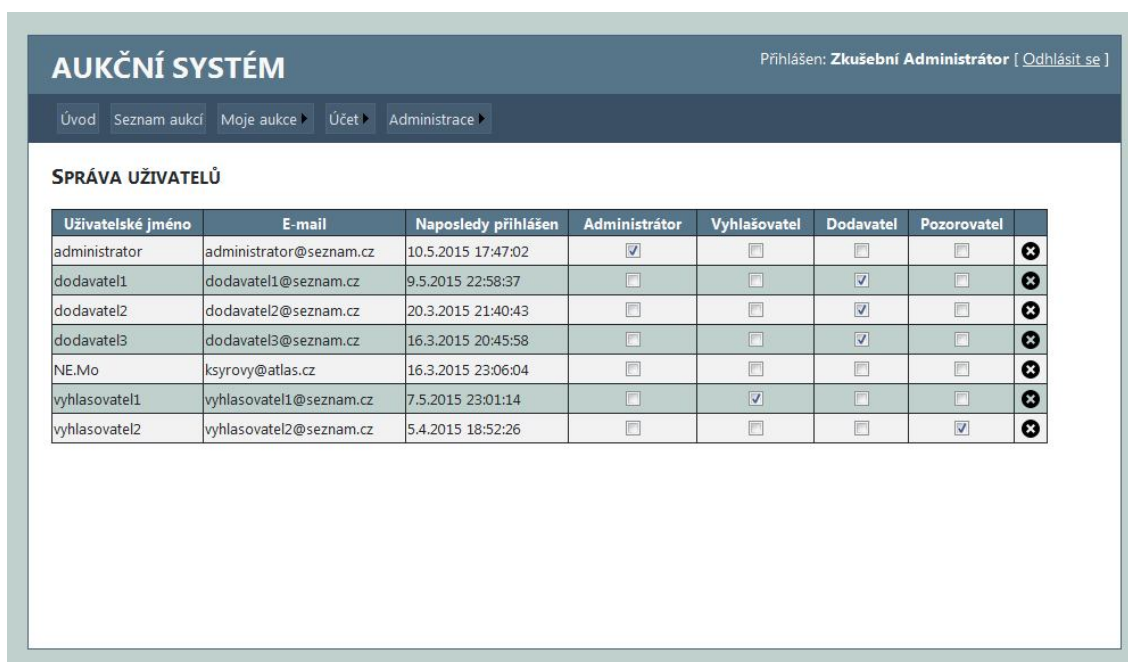
- ARLOW, J., NEUSTADT, I. *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky*. Vyd. 1. Brno: Computer Press, 2007, 568 s. ISBN 978-80-251-1503-9.
- BRAMBILLA, M., BUTTI, S. *Fifteen Years of Industrial Model-Driven Development in Software Front-ends: From WebML to WebRatio and IFML* [online]. Milán: Politecnico di Milano, 2013 [cit. 2015-04-10]. Dostupné z: <http://dbgroup.como.polimi.it/brambilla/sites/dbgroup.como.polimi.it/brambilla/files/referencematerials/WebML-IFML-WebRatio-Novatica-EN.pdf>
- BRAMBILLA, M., COMAI, S., FRATERALI, P., MATERA, M. *Designing Web Applications with WebML and WebRatio* [online]. Milán: Politecnico di Milano, 2013 [cit. 2015-03-02]. Dostupné z: <http://www.webml.org/webml/upload/ent5/1/Chapter%209%20-%20WebML.pdf>
- BRAMBILLA, M., FRATERALI, P. *Interaction Flow Modeling Language: Model-Driven UI Engineering of Web and Mobile Apps with IFML*. Burlington: Morgan Kaufmann Publishers, 2014, 422 p., ISBN 978-0-12-800108-0.
- BRIND, M., SPAANJAARS, I. *Beginning ASP.NET Web Pages with WebMatrix*. Indianapolis: John Wiley & Sons, 2011, 432 p. ISBN 978-1-118-05048-4.
- BUCHALCEVOVÁ, A., PAVLÍČKOVÁ, J., PAVLÍČEK, L. *Základy softwarového inženýrství - materiály ke cvičení*. 1. vyd. Praha: Vysoká škola ekonomická, 2007, 222 s. ISBN 987-80-245-1270-9.
- CAISON, CH., C. *ASP.NET Programmer's Reference*. 1st ed. New York City: McGraw-Hill Osborne Media, 2002, 350 p. ISBN 978-0-07-219049-6.
- CASTELEYN, S., DANIEL, F., DOLOG, P., MATERA, M. *Engineering Web Applications*. Heidelberg: Springer-Verlag Berlin, 2009, 349 p. ISBN 978-3-540-92200-1.
- CERI, S., FRATERALI, P., BONGIO, A., BRAMBILLA, M., COMAI, S., MATERA M. *Designing Data-Intensive Web Applications*. San Francisco: Morgan Kaufmann Publishers, 2003, 575 p. ISBN 1-55860-843-5.
- DE TROYER, O., CASTELEYN, S., PLESSERS, P. *Web Engineering: Modelling and Implementing Web Applications*. London: Springer-Verlag, 2008, 462 p. ISBN 978-1-84628-922-4.
- FINK, G., FLATOW I. *Pro Single Page Application Development: Using Backbone.js and ASP.NET*. 1st ed. New York City: Apress Media, 2014, 324 p. ISBN 978-1-4302-6673-0.
- GAYLORD, J., WENZ, CH., RASTOGI, P., MIRANDA, T., HANSELMAN, S. *Professional ASP.NET 4.5 in C# and VB*. Indianapolis: John Wiley & Sons, 2013, 1 449 p. ISBN 978-1-118-31182-0.
- HAVERBEKE, M. *Eloquent JavaScript: A Modern Introduction to Programming*. 2nd ed. San Francisco: No Starch Press, 2014, 472 p. ISBN 978-59327-584-6.

- IFML: The Interaction Flow Modeling Language*. In: Object Management Group, Inc. [online]. 2015 [cit. 2015-04-08]. Dostupné z: <http://www.ifml.org/>
- JAYANTY, S., S., K. *Microsoft SQL Server 2008 R2 Administration Cookbook*. 1st ed. Birmingham: Packt Publishing, 2011, 468 p. ISBN 978-1-84968-144-5.
- JORGENSEN, P., C. *Software Testing: A Craftsman's Approach*. 3rd ed. Boca Raton: Auerbach Publications, 2008, 440 p. ISBN 0-8493-7475-8.
- KOCH, N., KRAUS, A., HENNICKER, R. *The Authoring Process of the UML-based Web Engineering Approach*. In: First International Workshop on Web-Oriented Software Technology. 2001 [cit. 2015-04-20]. Dostupné z: <http://users.dsic.upv.es/~west/iwwost01/files/contributions/NoraKoch/Uwe.pdf>
- Migrating from WebML to IFML*. In: WebRatio [online]. 2013 [cit. 2015-02-080]. Dostupné z: <http://www.webratio.com/learn/learningobject/migrating-from-webml-to-ifml>
- MOLHANEC, M. *Metodologie orientované na tvorbu webových sídel*. In: Tvorba software 2003. Ostrava: Tanger, 2003, s. 134-144. ISBN 80-85988-83-6.
- MOLHANEC, M. *The Object-Oriented Hypermedia Design Model*. In: Objekty '2001. Praha: Česká zemědělská univerzita, 2001, s. 43-52. ISBN 80-213-0829-X.
- OBJECT MANAGEMENT GROUP, INC. *Interaction Flow Modeling Language* [online]. 2015 [cit. 2015-04-15]. Dostupné z: <http://www.omg.org/spec/IFML/1.0/PDF>.
- Rapid mobile App and Web application development Platform*. In: WebRatio [online]. 2015 [cit. 2015-04-27]. Dostupné z: <http://www.webratio.com/site/content/en/home>
- RYDVAL, S. *IFML – jazyk pro modelování interakcí aplikace s uživatele*. In: Můj život s UML. 2013 [cit. 2015-04-10]. Dostupné z: <http://blok.ocup.cz/2013/04/ifml-jazyk-pro-modelovani-interakci-gui.html>
- SCHWABE, D. *The Object-Oriented Hypermedia Design Model (OOHDM)* [online]. 2008 [cit. 2015-04-18]. Dostupné z: <http://www.oohdm.telemidia.puc-rio.br/>
- TŮMA, J. *Generování webového rozhraní*. In: WebČesky [online]. 2012 [cit. 2015-04-18]. Dostupné z: <http://www.webcesky.cz/generovani-weboveho-ui/>
- WAZLAWICK, R., S. *Object-Oriented Analysis and Design for Information Systems: Modeling with UML, OCL and IFML*. Waltham: Morgan Kaufmann Publisher, 2014, 439 p. ISBN 978-0-12-418673-6.
- ZELEŇKA, P. *WebML – datové modelování*. In: Interval [online]. 2004b [cit. 2015-04-15]. Dostupné z: <https://www.interval.cz/clanky/webml-datove-modelovani>
- ZELEŇKA, P. *WebML – proces vývoje webové aplikace (specifikace požadavků)*. In: Interval [online]. 2004a [cit. 2015-04-13]. Dostupné z: <https://www.interval.cz/clanky/webml-proces-vyvoje-webove-aplikace-specifikace-pozadavku/>

- ZELENKA, P. *WebML – projektování webových aplikací*. In: Interval [online]. 2003 [cit. 2015-03-02]. Dostupné z: <http://interval.cz/clanky/webml-projektovani-webovych-aplikaci/>
- ZELENKA, P. *WebML – tvorba aplikací v prostředí WebRatio*. In: Interval [online]. 2004c [cit. 2015-04-27]. Dostupné z: <https://www.interval.cz/clanky/webml-tvorba-aplikaci-v-prostredi-webratio/>

Přílohy

A Náhled jednotlivých částí webové aplikace



AUČNÍ SYSTÉM Přihlášen: Zkušební Administrátor [[Odhlásit se](#)]

Úvod Seznam aukcí Moje aukce Účet Administrace

SPRÁVA UŽIVATELŮ

Uživatelské jméno	E-mail	Naposledy přihlášen	Administrátor	Vyhlášovatel	Dodavatel	Pozorovatel	
administrator	administrator@seznam.cz	10.5.2015 17:47:02	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
dodavatel1	dodavatel1@seznam.cz	9.5.2015 22:58:37	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
dodavatel2	dodavatel2@seznam.cz	20.3.2015 21:40:43	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
dodavatel3	dodavatel3@seznam.cz	16.3.2015 20:45:58	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
NE.Mo	ksyrov@atlas.cz	16.3.2015 23:06:04	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
vyhlasovatel1	vyhlasovatel1@seznam.cz	7.5.2015 23:01:14	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
vyhlasovatel2	vyhlasovatel2@seznam.cz	5.4.2015 18:52:26	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Obr. 19 Správa uživatelů

AUKČNÍ SYSTÉM

Přihlášen: **Zkušební Administrátor** [[Odhlásit se](#)]

[Úvod](#) [Seznam aukcí](#) [Moje aukce](#) [Účet](#) [Administrace](#)

VLOŽIT NOVOU AUKCI

Aukce

Název:

Popis:

Začátek:

Konec:

Kategorie: Elektrická energie

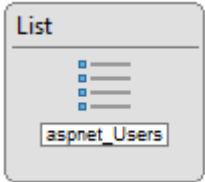
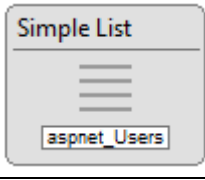
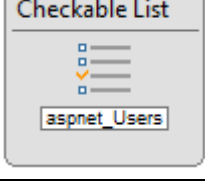
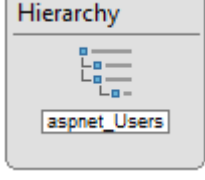
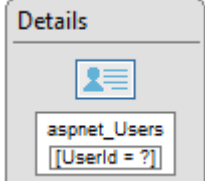
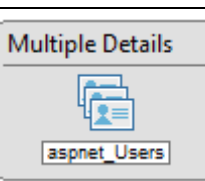

Zveřejnit aukci:

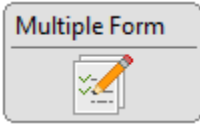



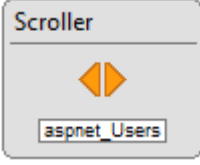
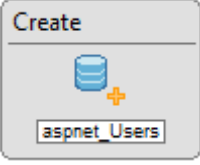
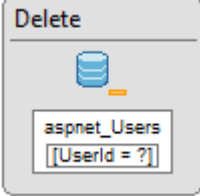
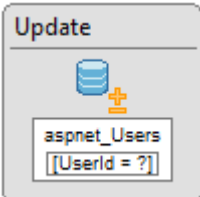

Aukce bude bude přístupná k podání nabídky přidáním dodavatelům v sekci Seznam aukcí.


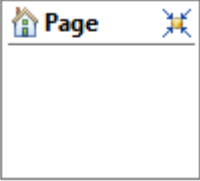
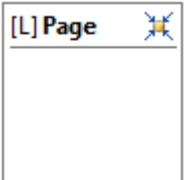
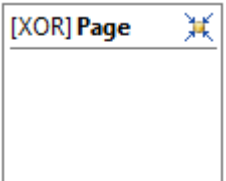

Obr. 20 Vytvořit novou aukci

B Seznam základních elementů IFML

Tab. 4 Seznam základních elementů IFML

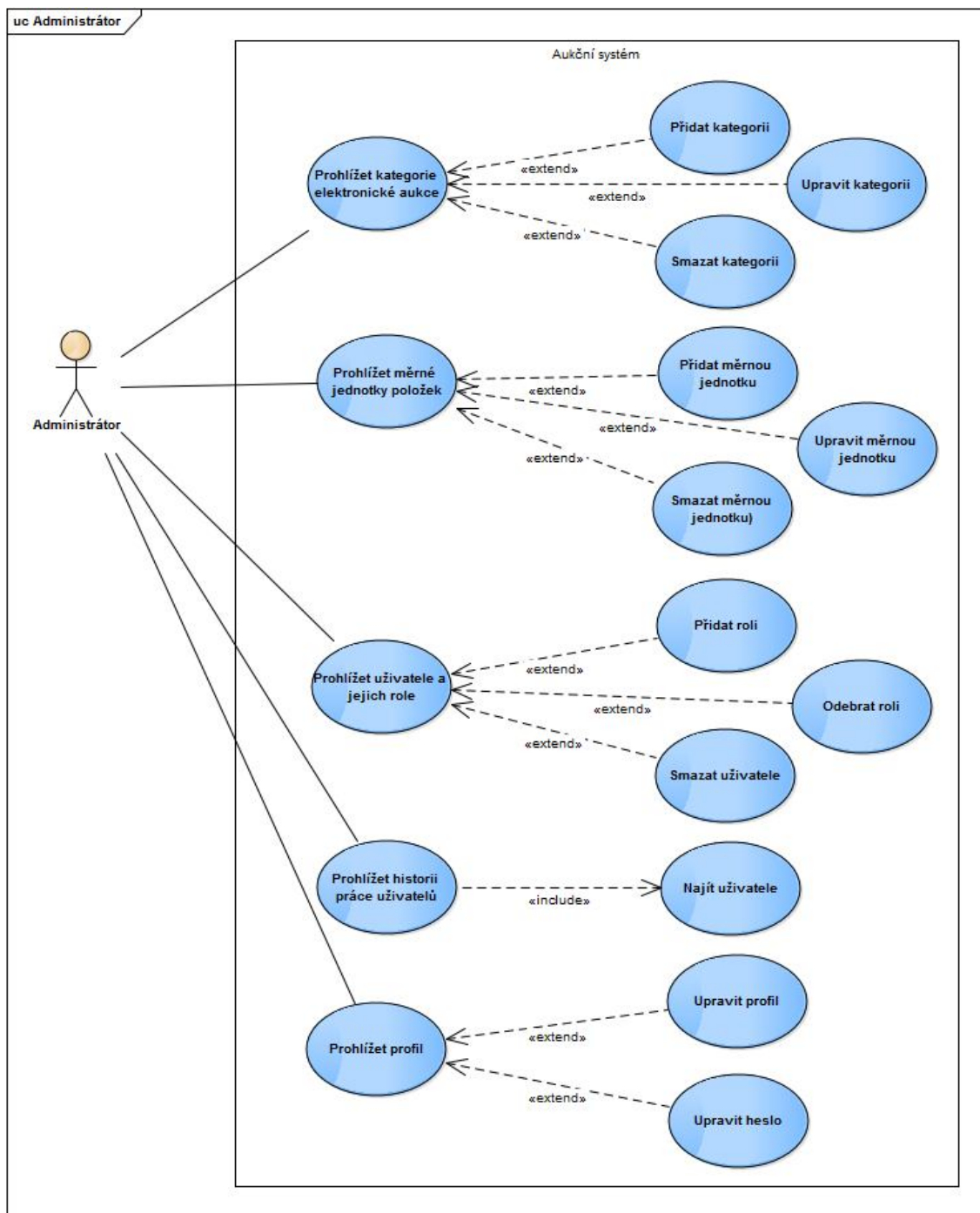
Pojem	Symbol	Význam
List		Jednoduchý seznam, který umožňuje posouvání a dynamické třídění.
Simple List		Představuje seznam instancí dané entity, umožňuje výběr jedné z instancí.
Checkable List		Seznam instancí dané entity, který umožňuje vícenásobný výběr.
Hierarchy		Instance, které jsou uspořádané hierarchicky do stromové struktury.
Details		Podrobné zobrazení informací jedné instance.
Multiple Details		Zobrazení informací o kolekci instancí jedné entity.
Form		Pole formuláře založené pro zadávání dat uživatelem.

Pojem	Symbol	Význam
Multiple Form		Formulář obsahující více polí pro zadání dat uživatelem.
Calendar		Komponenta kalendář pro výběr data, případně času.
Message		Komponenta umožňující odesílání zpráv.
Alphabet		Komponenta sloužící pro zobrazení obsahu stránky.
Scroller		Slouží pro procházení instancí entit rozdělených do listů.
Create		Vytvoří novou instanci a záznam přidá do databáze.
Delete		Odstranění jedné nebo více instancí vybrané entity.
Update		Aktualizace jednoho nebo více objektů stejné třídy.
Login		Přihlášení uživatele do systému, ověření identity.

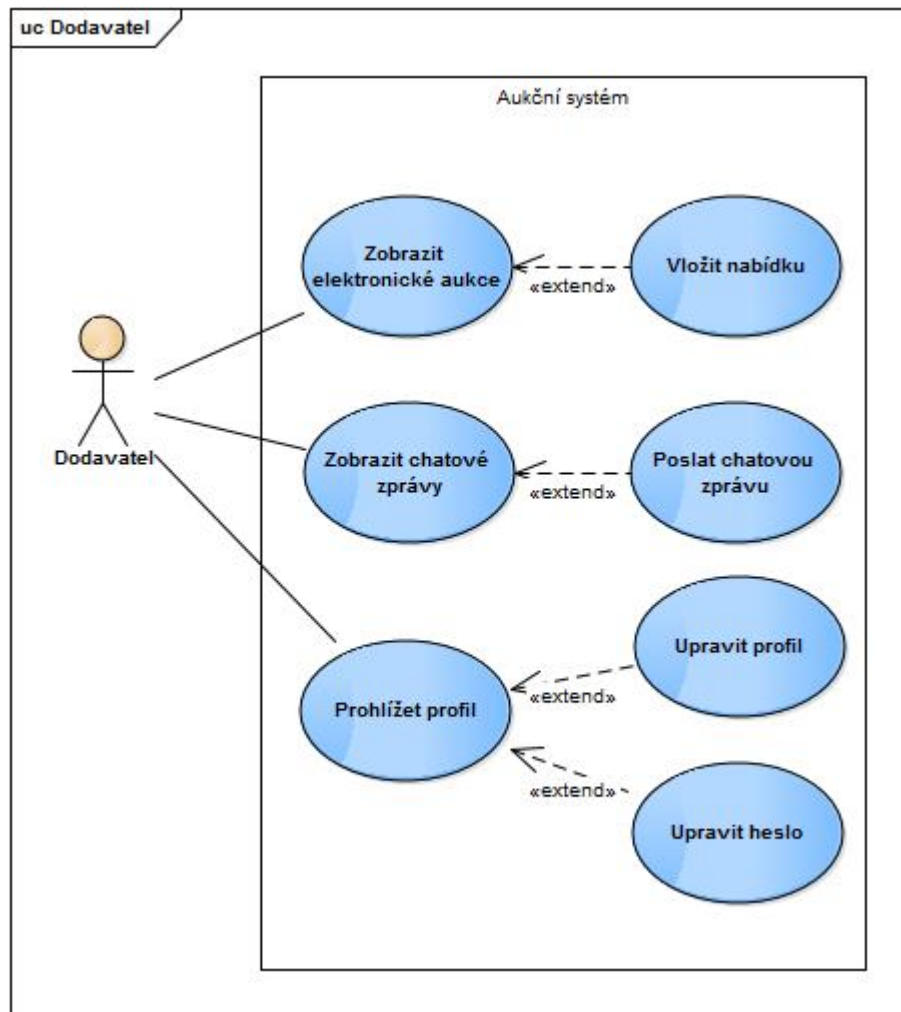
Pojem	Symbol	Význam
Logout		Odhlášení uživatele ze systému, opouští neveřejnou část.
Default		Určuje výchozí stránku pro určitou oblast.
Landmark		Určuje stránku, na kterou se lze dostat z jakékoliv jiné stránky.
XOR		Takto označená stránka v sobě zahrnuje vnořené stránky, na jejichž obsah se lze dostat z této stránky.
Navigation Flow		Určuje navigaci mezi jednotlivými stránkami.

Zdroj: Migrating from WebML to IFML, 2013.

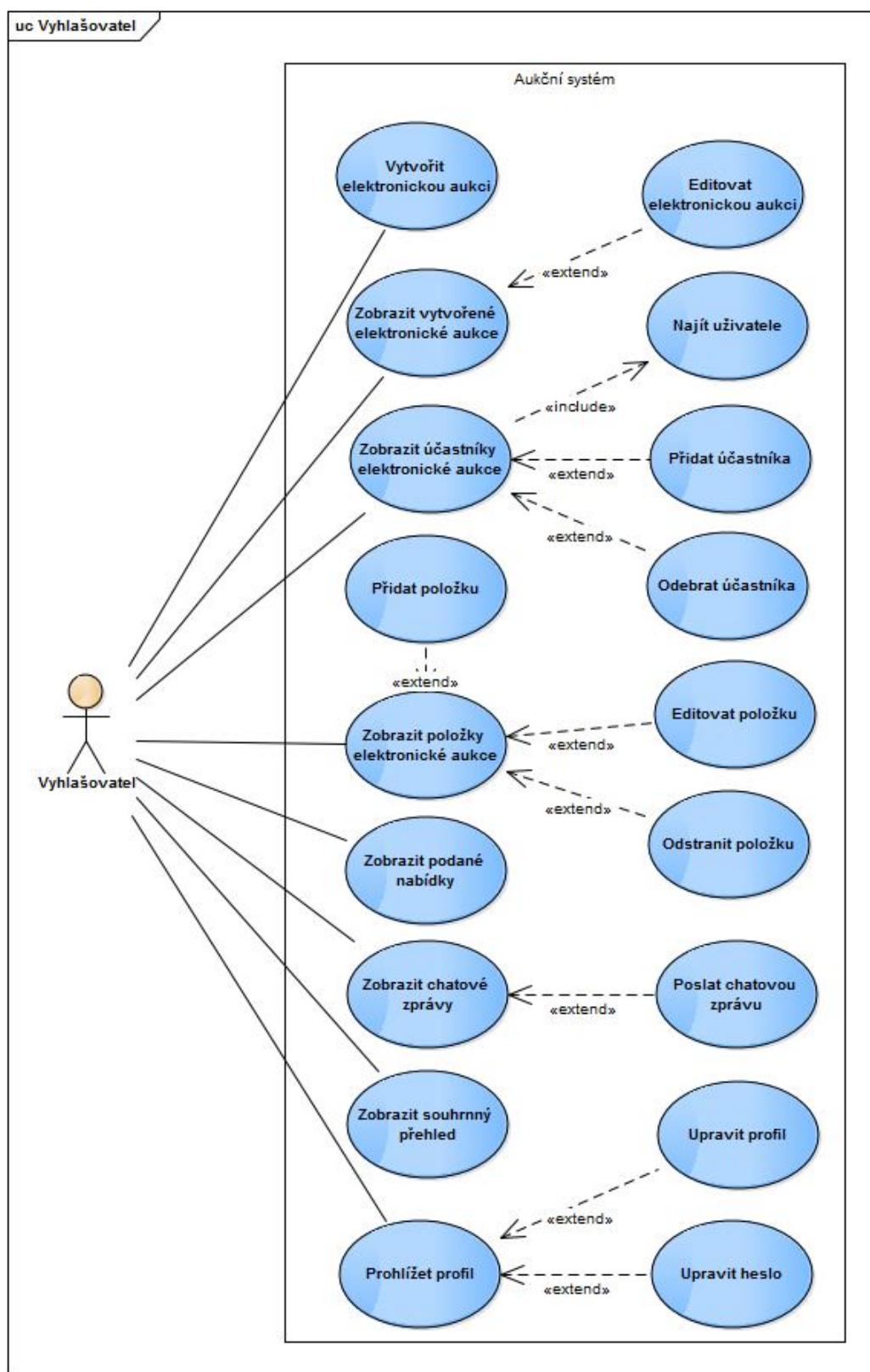
C Diagramy případů užití



Obr. 21 Diagram případů užití pro administrátora

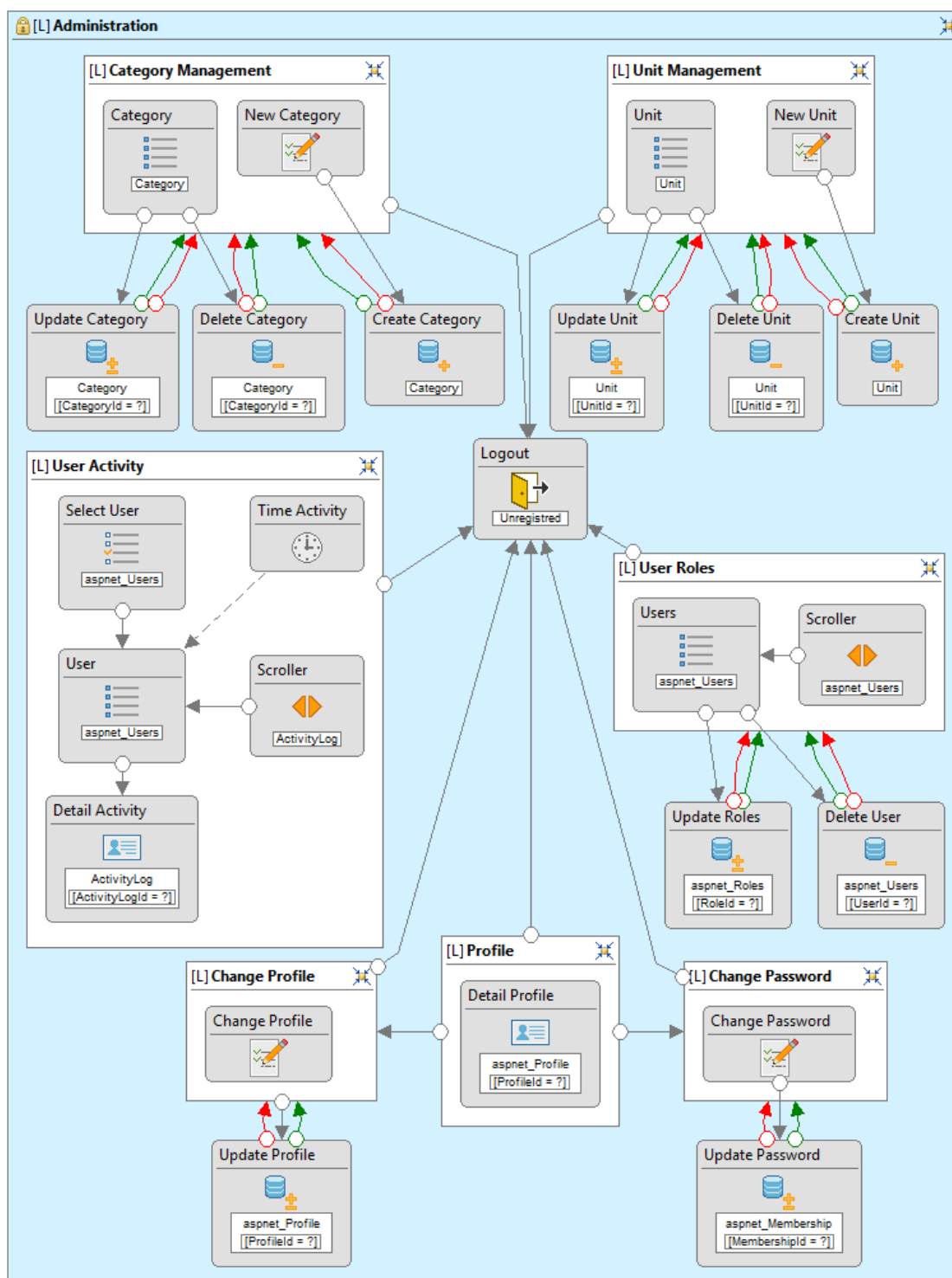


Obr. 22 Diagram případů užití pro dodavatele

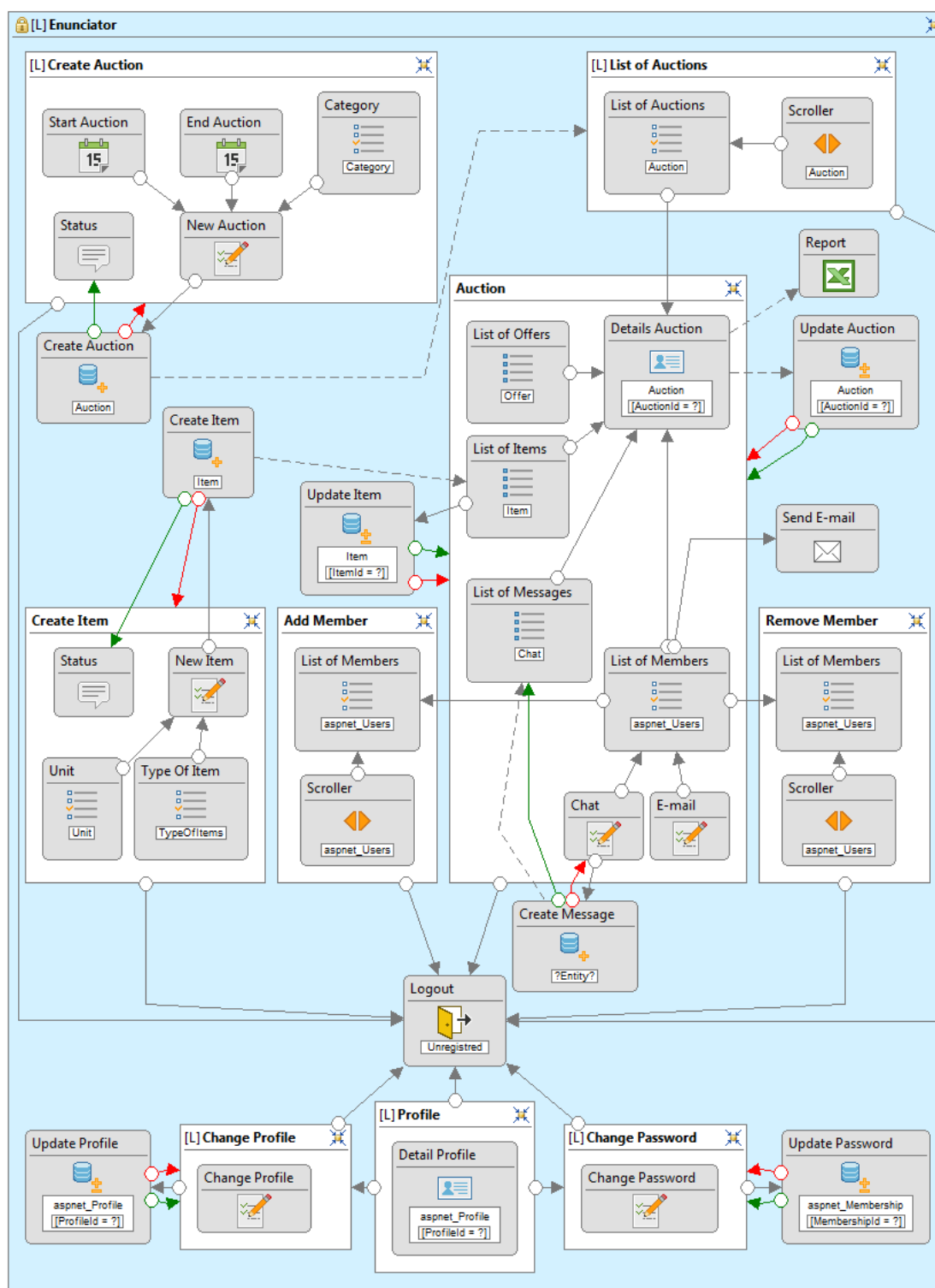


Obr. 23 Diagram případů užití pro vyhlášovatele

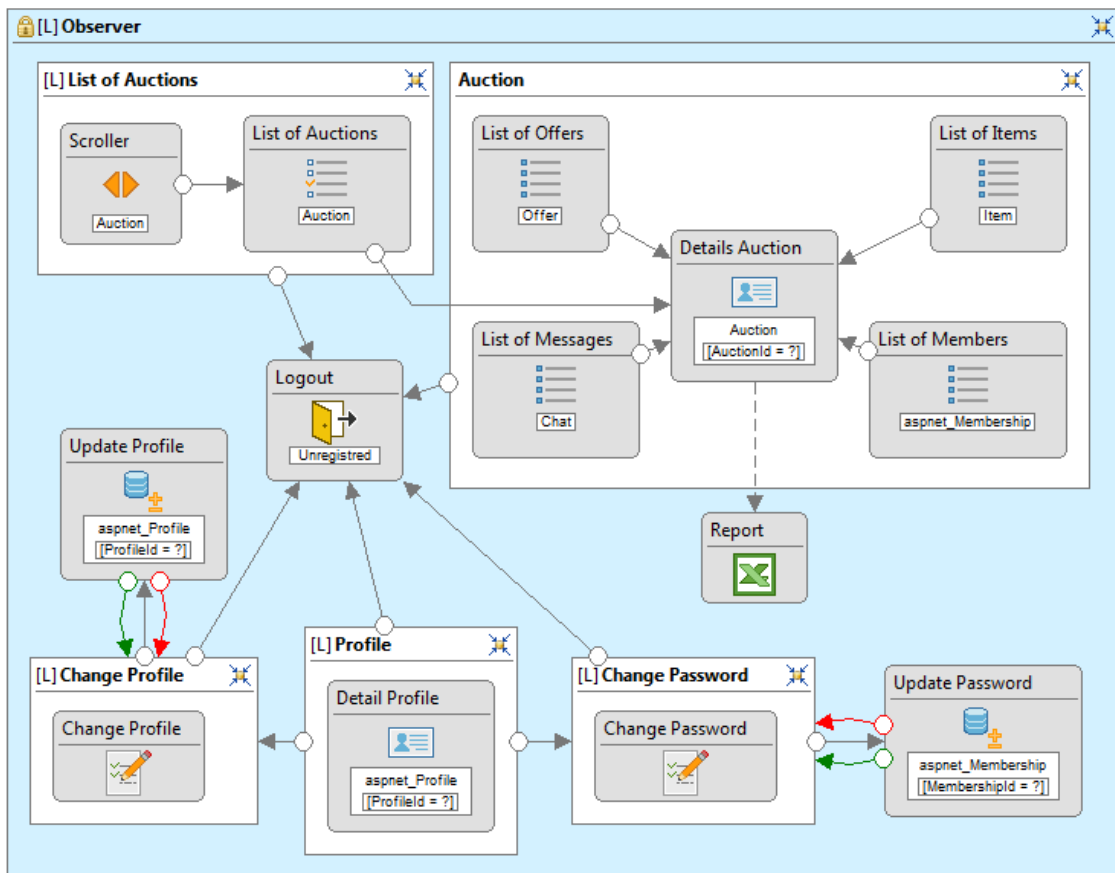
D IFML modely webové aplikace



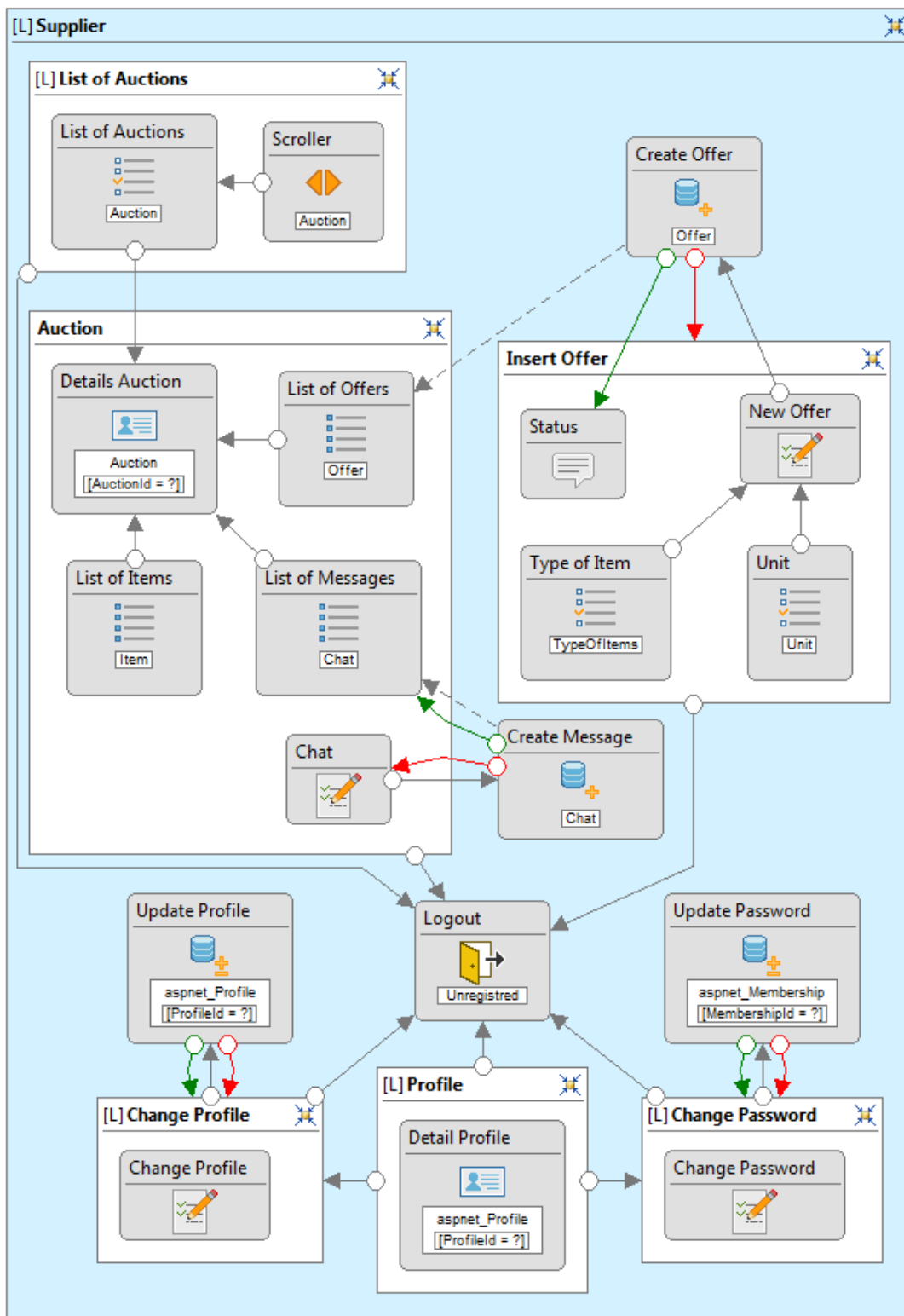
Obr. 24 IFML model pro administrátora webové aplikace



Obr. 25 IFML model pro vyhlášovatele elektronické aukce



Obr. 26 IFML model pro pozorovatele elektronické aukce



Obr. 27 IFML model pro dodavatele