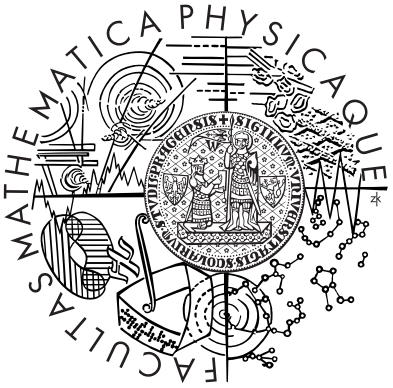
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

# Diplomová práce




Václav Koudelka

*Sémantická personalizace*

Katedra softwarového inženýrství
Vedoucí diplomové práce: RNDr. Filip Zavoral, Ph.D.

Studijní program: Informatika, Architektura a principy systémového prostředí

Děkuji vedoucímu diplomové práce RNDr. Filipu Zavoralovi, Ph.D. za odborné vedení práce a za poskytnutí cenných rad.

Prohlašuji, že jsem svou diplomovou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne 5. srpna 2009 Václav Koudelka

# Obsah

# Seznam obrázků

Název práce: *Sémantická personalizace*
Autor: *Václav Koudelka (vkoudelka@gmail.com)*
Katedra (ústav): *Katedra softwarového inženýrství*
Vedoucí diplomové práce: *RNDr. Filip Zavoral, Ph.D.*
e-mail vedoucího: *zavoral@ksi.mff.cuni.cz*
Abstrakt: Cílem této diplomové práce je navrhnout a implementovat nástroj v podobě frameworku, který bude určen pro webové tvůrce a měl by umožňovat metody personalizace v rámci libovolné webové aplikace. Framework bude získávat informace o provozu webové aplikace, bude se snažit zjistit, jak je aplikace využívána uživateli a na základě těchto poznatků se pokusí zefektivnit používání aplikace pomocí úprav prostředí. Důraz při návrhu frameworku přitom bude kladen na snadnost použití u běžných webových aplikací, ke své práci by měl používat prostředky běžně dostupné na jakémkoliv standardním webhostingu. Každému uživateli bude na základě jeho chování v aplikaci přizpůsobeno prostředí na míru. Součástí bude inteligentní rozhodovací mechanismus, jehož vstupem budou data o používání webové aplikace a data získaná od uživatele. Výstupem bude sada doporučení pro úpravu aplikace. Součástí práce bude také vzorová webová aplikace, ve které budou demonstrovány možnosti frameworku.
Klíčová slova: uživatelské modely, adaptivní systémy, web mining

Title: *Semantic personalization*
Author: *Václav Koudelka (vkoudelka@gmail.com)*
Department: *Department of Software Engineering*
Supervisor: *RNDr. Filip Zavoral, Ph.D.*
Supervisor's e-mail address: *zavoral@ksi.mff.cuni.cz*
Abstract: The aim of this work is to design and implement a tool in form of framework, which is destined for web architects and should enable methods of personalization within the scope of arbitrary web application. Framework will obtain information on functioning of web application, it will try to determine, how is application used by users and based on this knowledge it will attempt to make using of application more efficient by means of environmental adaptations. During framework designing emphasis will be put on easiness of using within common web application, to its work it should use only means commonly available within any standard webhosting. For every user will be application environment personalized on the basis of his behaviour. Part of framework will be intelligent decision mechanism, whose input will be data about web application usage and data retrieved from user. Output will be set of recommendations for application modification. Part of thesis is also sample web application, in which framework potential is demonstrated.
Keywords: user modeling, adaptive systems, web mining

# 1 Úvod

V mnoha současných webových aplikacích vzniká potřeba přizpůsobovat služby aplikace uživatelům, tento proces přibližování služeb webových aplikací a potřeb uživatelů se nazývá personalizace. Důvody pro personalizaci mohou být různé, nicméně setkat se s ní lze u aplikací v komerční sféře, v portálech statní správy, v univerzitních portálech i jinde. Jeden důvod pro její použití je shodný u komerčních i u nekomerčních aplikací. Množství dat na webu je obrovské, hledání konkrétních informací a navigace se stali obtížnými často i v rámci jednoho portálu. Personalizace může navigaci usnadnit a případně nabídnout uživateli co možná nejvhodnější prezentaci informací. U komerčních společností obvykle bývá přístup k zákazníkům prioritou, zástupci společnosti chtějí, aby zákazník cítil, že společnost k němu přistupuje jako k jednotlivci a že pro ně není pouze jedním z davu [5]. Podobně i v portálech státní správy [1] je cílem nabídnout návštěvníkovi osobnější přístup a usnadnit mu komunikaci s úřadem poskytnutím informací a služeb, které jsou pro něj užitečné.

Současné způsoby personalizace ve webových aplikacích jsou většinou poměrně jednoduché. Personalizace se často objevuje v komerčních aplikacích a je založena na základě obchodních požadavků či zkušenosti s prodejem. Typickým příkladem je nabídka podobného zboží u detailu produktu v různých elektronických obchodech. U některých aplikací je uživateli umožněno nastavit si svůj vlastní profil a na základě vyplněných dat dochází k úpravě prezentace informací. Několik ukázek složitějších přístupů k personalizaci je zmíněno v sekci související práce. Většinou to bývají poměrně komplikované přístupy k personalizaci aplikací, které jsou šité na míru do určitého prostředí, např. [1] se zabývá personalizací portálů statní správy.

## 1.1 Cíl

Cílem této diplomové práce je navrhnout a implementovat nástroj v podobě frameworku, který bude určen pro webové tvůrce a měl by umožňovat metody personalizace v rámci libovolné webové aplikace. Framework bude získávat informace o provozu webové aplikace, bude se snažit zjistit, jak je aplikace využívána uživa-

teli a na základě těchto poznatků se pokusí zefektivnit používání aplikace pomocí úprav prostředí. Důraz při návrhu frameworku přitom bude kladen na snadnost použití u běžných webových aplikací, ke své práci by měl používat prostředky běžně dostupné na jakémkoliv standardním webhostingu. Každému uživateli bude na základě jeho chování v aplikaci přizpůsobeno prostředí na míru. Součástí bude inteligentní rozhodovací mechanismus, jehož vstupem budou data o používání webové aplikace a data získaná od uživatele. Výstupem bude sada doporučení pro úpravu aplikace.

Součástí práce bude také vzorová webová aplikace, ve které budou demonstrovány možnosti frameworku.

## 1.2 Související práce

### 1.2.1 AVANTI

AVANTI [13] je systém, který funguje jako webový portál města, zprostředkovává informace o veřejných službách, dopravě, budovách, atd.. Je určen pro širokou škálu uživatelů od občanů města po turisty. Cílem je nabídnout návštěvníkovi informace, které chce, v podobě, která mu nejvíce vyhovuje. V systému je pamatováno na handicapované uživatele.

Každý uživatel přistupující do systému je podroben vstupnímu interview, pomocí kterého je zařazen do kategorie (kategorií). Další informace o u uživateli se sbírají během jeho interakce s portálem, sledují se stránky, které navštěvuje a systém se z nich snaží vydedukovat zájmy uživatele. Adaptace jsou realizovány pomocí tzv. stereotypů. Stereotyp je zde definován jako soubor domněnek o určité skupině uživatelů. Domněnky se týkají zájmů uživatelů neboli objektů, které by měly mít pro uživatele nějaký přínos. Pokud uživatel splňuje podmínky členství v nějaké skupině, k níž je přiřazen nějaký stereotyp, pak stereotyp je aktivován a uživateli jsou nabídnuty objekty z tohoto stereotypu.

Adaptace na jednotlivých stránkách jsou umožněny přidáním speciálních tagů do HTML kódu stránek, které pak systém AVANTI vyhodnotí. Speciální tagy obsahují adaptace a podmínky, které musí být splněny, aby adaptace byla provedena. Pro definování podmínek je použit skript uložený v dalším souboru.

### 1.2.2 AHA

Adaptive Hypermedia Architecture (AHA) [14] je systém vyvinutý na Eindhoven University of Technology. Tento systém je navržen tak, aby umožňoval adaptivitu webových aplikací obecně. Autoři zmiňují pět typů webových aplikací, pro které je systém určen: on-line informační systémy, on-line help, naučná hypermedia, institucionální hypermedia a osobní hypermedia.

Uživatelský model je zde reprezentován sadou "konceptů", které mají booleovské hodnoty. Koncept může být například *uživatel četl informace XY*.

Adaptace jsou podobně jako v AVANTI umožněny speciálními tagy v HTML kódu. Definice adaptací jsou složeny z podmínky a textu (HTML kódu), který je zobrazen, pokud je podmínka splněna. Podmínka je výčet konceptů, pokud jsou všechny koncepty v podmínce true, tak je adaptace provedena.

### 1.2.3 FIT

Projekt FIT [15] je self-adaptive framework určený pro webové portály státní správy. Cílem je zvýšit kvalitu elektronických služeb státní správy tak, aby lépe vyhovovala jednotlivým uživatelům. Prostředkem je automatická úprava prostředí podle záměrů a charakteristik uživatelů, ale i organizací služeb v pozadí portálu.

FIT je poměrně robustní projekt, je založený na používání ontologií. Představuje FIT Ontologii, která modeluje vlastnosti mezi uživateli, administrativními procesy a portály. FIT Ontologie je klíčová pro provádění adaptací.

FIT framework je složen z několika částí (modulů), které se starají o jednotlivé aktivity frameworku. Jsou to například modul pro logování, modul pro datamining, modul pro adaptace, atd.. Tyto moduly mezi sebou mají definované komunikační rozhraní. Jak vypadají výstupy jednotlivých modulů určuje FIT ontologie. Rovněž pomocí ontologie je definována struktura aplikace a popis vlastností uživatelů. V procesu adaptace jsou mimo jiné ontologií definovány koncepty uživatel, cíl uživatele, chování uživatele, kategorie uživatele, pravidlo, adaptace. Dále je definováno jejich propojení. Každý uživatel má nějaký cíl a úkolem jednoho z modulů FIT frameworku je tento cíl zjistit. Výchozím bodem pro zjišťování cíle uživatele je jeho chování. Na základě jeho chování se také určuje kategorie uživatele. V systému, který FIT framework používá, jsou definováné adaptace i cíle uživatelů (co může

uživatel v systému dělat), každá adaptace je napojena na jeden nebo více cílů, což znamená, že adaptace usnadní dosažení tohoto cíle. Pokud framework zjistí jaký je cíl nějakého uživatele, tak aplikuje adaptace, které jsou s tímto cílem spojeny. Další možnost aplikování adaptací je pomocí pravidel. Pravidlo je propojení kategorie uživatele s adaptacemi, určuje adaptace, které jsou vhodné pro určité kategorie uživatelů.

Samotnou adaptaci pak provádí samostatný modul. Po aplikování adaptace se ještě zjišťuje, zda adaptace měla pozitivní efekt.

# 2 Rozbor problematiky

Při vytváření samostatného inteligentního rozhodovacího mechanismu pro samostatnou adaptaci webové aplikace jsou v zásadě tři otázky, se kterými je nutné se vypořádat.

1. na základě jakých vstupních dat rozhodování provádět
2. z jaké množiny adaptací vybírat
3. kterou adaptaci udělat v jakou chvíli

V následujících částech se budu trochu podrobněji věnovat třem zmíněným problémům, jejich popisu a možnostem jejich řešení v rámci frameworku.

## 2.1 Vstupní data

Pro rozhodovací mechanismy obecně platí, že čím přesnější budou informace na základě kterých se rozhodování provádí, tím větší je pravděpodobnost správného výstupu. Proto pro framework je získávání dat klíčovou otázkou. Požadavek je, aby data byla ve strojově čitelném formátu, aby co nejlépe reflektovala realitu a aby se získávala pouze ta data, která mají pro rozhodování frameworku smysl.

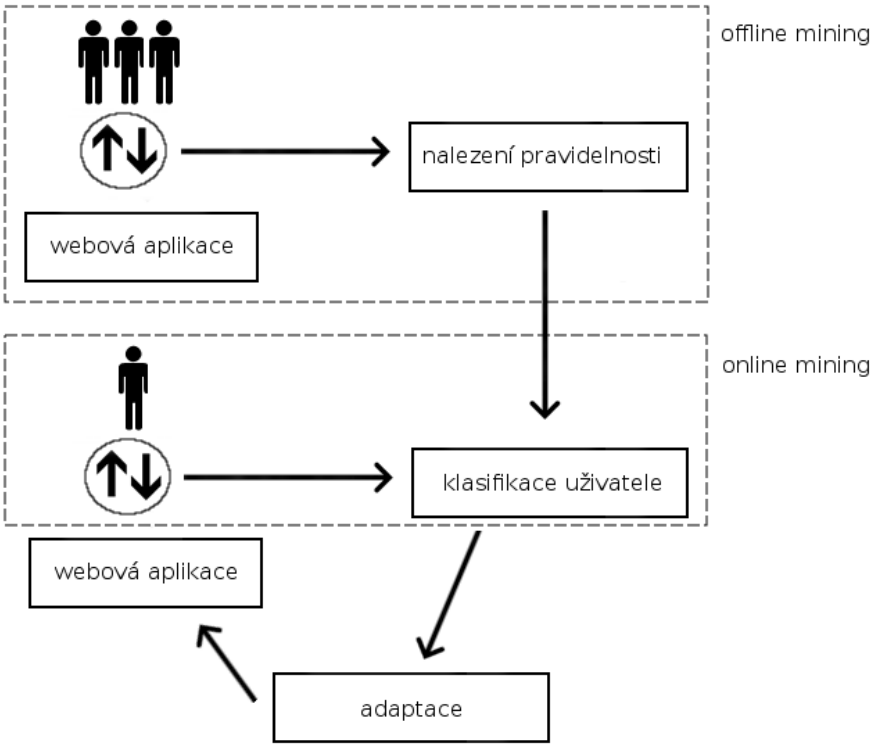
Metody zabývající se získáváním dat z webových aplikací jsou známy pod pojmem web mining. Základní kategorie web mining jsou web content mining, web structure mining a web usage mining [3].

[1] rozděluje web mining na offline mining a online mining. Offline mining pracuje například se systémovými logy, kde zkoumá aktivity uživatelů a hledá v nich opakující se vzory chování (obr. 2.1). Online mining je použito ve chvíli, kdy webovou aplikaci navštíví uživatel, pozoruje pouze jeho chování a úkolem je identifikovat v záznamu jeho aktivity schémata objevené při offline mining.

Offline mining nemá přímý vliv na proces adaptování webové aplikace, provádění offline mining nespustí žádné adaptace. Je to samostatný proces, který může fungovat nezávisle na ostatních částech frameworku. Online mining je naopak úzce napojené na adaptace. Poznatky zjištěné při online mining jsou v reálném čase aplikovány při provádění adaptací.

[3] dále doporučuje proces získávání dat z webu rozdělit do čtyř podúloh:

1. získání zdrojů




Obr. 2.1: offline/online mining

2. předzpracování dat
3. zobecnění (nalezení obecných vzorových modelů v získaných datech)
4. interpretace získaných modelů

### 2.1.1 Získání zdrojů

#### Web content mining

Web content, neboli obsah webu, je spojení několika typů dat jako text, obraz, audio či video. Web content mining je proces, který nad těmito daty hledá výskyty společných rysů (u textových dat je to výskyt stejných slov nebo slovních spojení). Současný výzkum v oblasti web content mining klade důraz na textová data.

Web content mining pro adaptaci webových aplikací je přínosný z pohledu získávání informací o struktuře jednotlivých stránek aplikace. Samotná textová data zůstanou v první fázi návrhu frameworku stranou. Cílem je hledání společných rysů v chování uživatelů a na základě toho určovat objekty ve webové aplikaci, o které by mohli mít zájem, nezávisle na tom, co obsahují.

Znalost informací o struktuře jednotlivých stránek a o tom, které objekty se na stránkách vyskytují, doplňují informace o chování uživatelů aplikace. Samotný záznam historie uživatele nemusí být vždy vypovídající. Uvažujme příklad získání cesty uživatele po webu z logu serveru. Ten ukáže, která url uživatel navštívil, což je informace jistě zajímavá, ale další, leckdy podstatnější informace, zůstanou skryty. Zaznamenaná url nemusí mít na první pohled nic společného, ale prozkoumání jejich obsahu může ukázat, že v těchto stránkách dochází k častému výskytu určitých specifických objektů, které jsou předmětem uživatelova zájmu. Klasifikace stránek je spolu s informacemi o topologii webové aplikace důležitým faktorem pro pochopení a správné prozkoumání chování uživatelů.
V [6] se jednotlivé stránky dělí na pět hlavních typů:

- hlavní stránka - stránka, kterou uživatel navštíví první
- obsahová stránka - zde jsou obsahové informace, které aplikace nabízí
- navigační stránka - poskytuje odkazy na další stránky
- slovníková (look-up) stránka - poskytuje vysvětlení pojmů
- osobní stránka - obsahuje informace o osobách spojených s webovou aplikací

Dále se v aplikaci mohou vyskytovat stránky, které budou moci být zařazeny do více typů, například hlavní stránka je velmi často zároveň navigační. Některé stránky nemusí patřit do žádného ze základních typů, mohou to být různé specifické stránky, například formuláře. Zařazení konkrétních stránek do těchto typů je možné dělat buď manuálně nebo automaticky. První možnost je v režii tvůrce webu a spočívá v tom, že do kódu stránky bude přidán tag určující zařazení stránky. Automatická klasifikace stránek vychází z jejich fyzických vlastností. Například navigační stránka obsahuje málo textu, hodně odkazů a uživatel na ní stráví relativně krátkou dobu. Podrobněji se o fyzických vlastnostech jednotlivých typů stránek zmiňuje [6], který pro automatickou klasifikaci doporučuje algoritmus C4.5 [11].

Vzhledem k tomu, že framework bude pracovat s webovými aplikacemi obecně, měl by tvůrce webu mít možnost ke každé stránce (případně i k objektům na stránce) přidat vlastní metadata, která budou brána v potaz při zkoumání chování uživatelů. Vhodná metadata můžou být třeba klíčová slova vztahující se ke stránce nebo vlastnosti produktu, který je na stránce nabízen či představován.

#### Web structure mining

Každá webová aplikace je složena ze stránek (uzlů), které jsou mezi sebou propojeny odkazy. Web structure mining se snaží rozpoznat strukturu stránek a odkazů mezi nimi. Struktura webu v podstatě reflektuje to, jak tvůrce webu chtěl, aby byla stránka používána [6]. Každý odkaz existuje proto, že tvůrce chtěl, aby stránky, mezi kterými odkaz vede, byly spojeny.

#### Web usage mining

Web usage mining je automatické objevování a analýza vzorů v datech posbíraných nebo vygenerovaných během interakce uživatele s webovou aplikací neboli relace [2]. Relací je myšlena jedna návštěva webové aplikace. Relace je popsána akcemi (transakcemi), které uživatel během relace provedl. Relace je reprezentována transakční historií, což je zázam akcí uživatele, kde každá akce je opatřena časovým razítkem. Cílem je získání vzorových modelů uživatelů webové stránky. Vzorový model je obvykle soubor stránek a objektů, ke kterým je často přistupáno uživateli

v rámci určité skupiny uživatelů. Skupinou uživatelů se rozumí uživatelé, kteří mají společné zájmy, potřeby nebo cíle.

Nejčastějším zdrojem dat pro web usage mining je log http serveru. Informace poskytnuté server logem můžou zrekonstruovat pohyb uživatele po stránkách webové aplikace. Web server zaznamenává každý http požadavek od klienta do souborů a mezi informacemi, které ukládá, jsou obvykle [7]:

- IP adresa klienta
- datum a čas požadavku
- příkaz požadavku
- návratový kód http serveru určený pro klienta

Logování http serveru má několik nedostatků. Častý problém je ukládání stránek do cache na straně klienta nebo na proxy serveru. Typickým příkladem použití cache je stisknutí tlačítka "Zpět" v browseru, kdy browser neposílá http požadavek pro stránku, na kterou se vrací. Důsledkem je, že v získané cestě pohybu určitého uživatele po webové aplikaci můžou některé uzly chybět, to je možné řešit buď doplněním cesty na základě informace o struktuře webu, nebo použitím dalších prostředků web usage mining. Možnostem řešení tohoto problému se podrobněji věnuje [6]. Dalším nedostatkem server logu je identifikace unikátního uživatele. Log poskytuje informace o IP adresa klienta, ale ta může patřit proxy serveru, za kterým je více uživatelů [12]. Pro dodatečné ověření identifikace uživatele lze použít cookies.

Jak už bylo řečeno, server zaznamenává http požadavky na stránky a soubory, ale o tom, co se děje mezi jednotlivými požadavky, informace nemá. U některých aplikací (např. DHTML aplikace) tak akce uživatelů, které by potenciálně mohly být významné, zůstávají neobjeveny.

Další možností získávání dat o tom, jak je webová aplikace používána, jsou nástroje pro sledování chování uživatelů na straně klienta pomocí metod DHTML (např. Google Analytics [8]). Tyto nástroje mohou být poskytovány třetí stranou, jejich použití vyžaduje modifikaci každé stránky tak, aby obsahovala kus javascript kódu, který nástroj pro sběr dat inicializuje. Informace získané tímto způsobem poskytnou komplexní pohled na chování uživatele v rámci aplikace, co se týče jeho pohybu po stránkách, tak i aktivity v rámci jednotlivých stránek. Obvykle bývají zaznamenávány i informace získané z browseru.

Poslední relevantní zdroj dat je sám uživatel. Mnoho webových aplikací se snaží získat od uživatelů prostřednictvím různých formulářů nebo jiných aktivních prvků (např. hodnocení produktů). Takto získané informace mohou být významné, jako třeba uvedený příklad hodnocení produktů. Tímto způsobem se zjistí, jaké produkty jsou pro uživatele zajímavé a uživatel bude moci být na základě poměrně dobře klasifikován do určité skupiny uživatelů. Podle [9] je ovšem tento způsob získávání dat problematický, protože naráží na nezájem uživatelů. Uživatelé neradi vyvíjejí snahu navíc, i když vědí, že v dlouhodobém měřítku by z ní profitovali.

### 2.1.2 Předzpracování dat

Předzpracování neboli preprocessing dat je soubor úloh, které se provedou nad získanými daty z logů serveru a informacích o obsahu a struktuře webové aplikace. Hlavními úkoly [6] [9] předzpracování jsou:

- vyčistění dat
- identifikace uživatelů
- určení relací uživatelů
- integrace informací o chování uživatele s informacemi o struktuře a obsahu aplikace
- spojit data z různých zdrojů
- formátování

#### Čištění dat

Objem získaných dat z logů serveru je obvykle velký a ne všechna data jsou užitečná. V přístupovém logu serveru jsou zaznamenány i http požadavky na soubory, které si uživatel explicitně nevyžádal. Jsou to například obrázky na stránce, dodatečné informace o vzhledu stránky a podobně. Požadavky na tyto pomocné soubory jsou při objevování vzorů v chování uživatelů nezajímavé, neboť patří do jedné transakce, která je reprezentována požadavkem na konkrétní stránku aplikace.

Čištění dat by se mělo týkat také záznamů v logu, které byly vytvořeny vyhledávacími roboty. Identifikace robotů bude provedena buď porovnáním se seznamem známých robotů, nebo alternativně podle toho, že robot jako první přistupuje k souboru "robots.txt" v kořenovém adresáři.

### Identifikace uživatelů

Identifikací uživatelů se rozumí přiřazení každého záznamu v logu, ke konkrétnímu uživateli, který aplikaci používá. To může být poměrně komplikované, jak bylo zmíněno v kapitole 2.1.1 IP adresa nemusí být vždy spolehlivý identifikátor. [6] popisuje možnost rozlišování uživatelů se stejnými IP adresami pomocí dodatečných informací z logu jako typ browseru klienta a dále pomocí heuristiky, která využívá topologii webové aplikace a zjišťuje, kam se uživatel z navštívené stránky mohl dostat. Variantou k identifikaci IP adresou je identifikace pomocí cookies, ani tato technika však není naprosto spolehlivá. Ne všechny webové aplikace používají cookies a pokud ano, můžou být cookies blokovány na straně klienta.

### Určení relací uživatelů

Relace (session) uživatele je definována jako jedna návštěva webové aplikace [9]. Soubory záznamů akcí během jedné relace jednoho uživatele by měly být výsledkem tohoto kroku. Uživatel může navštěvovat stránku opakovaně a analýza by s každou návštěvou měla nakládat samostatně. V [6] se jednotlivé návštěvy odlišují na základě časového limitu. V praxi to znamená, že pokud uživatel během určité doby neprovedl žádnou akci, tak jeho návštěva končí a příští použití webové aplikace bude bráno jako nová návštěva.

### Integrace informací o chování uživatele s informacemi o struktuře a obsahu aplikace

V tomto kroku budou spojeny poznatky získané jednotlivými metodami web mining. Ke stránkám nebo objektům, které se objevují v relaci, uživatele budou přidány informace zjištěné o těchto objektech pomocí web content mining a web structure mining. Pokud je potřeba, tak bude v tomto kroku provedeno doplnění cesty. Jak bylo zmíněno, v přístupovém logu http serveru nemusí být všechny záznamy, které by tam být měly. Doplnění cesty je metoda, která se snaží cestu zkompletovat. [6] popisuje způsob, jak doplnění provést v praxi, vychází z informací o topologii webu.

#### Spojení dat z různých zdrojů

Jak bylo zmíněno, data o chování uživatelů mohou být získávána různými způsoby. Sjednocení získaných dat do jednoho souboru je dalším krokem předzpracování dat. Po spojení by výsledek měl být jeden soubor s akcemi uživatelů a kde každý záznam je opatřen časovým údajem.

#### Formátování

Posledním krokem předzpracování dat je upravení výstupu tak, aby vyhovovalo algoritmu pro data mining, který bude použit.

### 2.1.3 Nalezení obecných vzorových modelů

Hledání vzorů a pravidelností ve velkém množství dat, které by v předchozích krocích mělo vzniknout, bude úlohou technik dobývání znalostí. Cílem je nalezení takových pravidel, která budou mít vypovídací hodnotu o tom, jak je webová aplikace používana a pomocí kterých budou moci být uživatelé klasifikováni do skupin podle jejich chování. Pro dobývání znalostí lze použít následující metody (výčet je převzán z [1]):

- analýza cesty (path analysis)
- asociační pravidla (association rules)
- sekvenční vzory (sequential patterns)
- clustering a klasifikace (clustering and classification)
- kolaborativní filtrování (collaborative filtering)

#### Path analysis

Tato metoda využívá topologii webové aplikace. Aplikaci reprezentuje jako graf, tak, že uzly jsou jednotlivé stránky a hrany jsou odkazy vedoucí mezi stránkami. Chování každého návštěvníka odpovídá cestě v grafu. Smyslem této metody je najít cesty, které návštěvníci často používají.

#### Association rules

Uživatelé webové aplikace používají určité stránky a objekty, které se v nich nacházejí. Metoda objevování asociačních pravidel hledá stránky a objekty často se vyskytující v jedné uživatelské relaci. Výstupem z této metody je například pravidlo typu "90% uživatelů, kteří navštívili stránku A, navštívili také stránku B".

#### Sequential patterns

Sequential patterns je technika, která hledá v uživatelských relacích často se opakující posloupnosti prováděných akcí. Funguje podobně jako hledání asociačních pravidel, bere ovšem v úvahu časovou chronologii provedených akcí.

#### Clustering and classification

Tato metoda se snaží do skupin zařadit uživatele, kteří jsou si navzájem podobní. Podobnost je možné určovat z mnoha ohledů, podle věku, rychlosti připojení, dalších atributů pokud jsou známé, případně podle stejného způsobu používání webové aplikace.

#### Collaborative filtering

Collaborative filtering je metoda, která předpovídá zájmy uživatele na základě sběru dat od předchozích uživatelů. Akce, které uživatel provedl, porovnává s akcemi uživatelů v minulosti a v případě, že najde uživatele, který prováděl stejné akce, zjistí, které další jsou u něj zaznamenány a ty jsou výdledkem predikce pro aktuálního uživatele. Akcí může být například zobrazení detailu produktu, stažení souboru, ohodnocení produktu, atd.

## 2.2 Sada adaptací

V první části této kapitoly uvedu možné typy adaptací webové aplikace a v druhé části se budu zabývat možnostem začleňování adaptací do práce frameworku.

### 2.2.1 Přehled adaptací

Adaptace webových aplikací se obvykle dělí do tří skupin na základě toho, čeho se adaptace týká. Jsou to adaptace obsahu, způsobu prezentace a struktury [1] [10].

#### Adaptace obsahu

Volitelné vysvětlení je určené pro uživatele, kteří nemají dostatečnou znalost o problematice, které se obsah na stránce věnuje. Toto vysvětlení bude užitečné pro uživatele, kteří se v problematice nevyznají, ale zbytečné pro uživatele, kteří znalost mají.

Volitelné přidání/odstranění detailních informací. Uživatelé zajímající se o téma, kterého se týká obsah na stránce mohou zajímat dodatečné informace, jež za normálních okolností nejsou zobrazovány.

Osobní doporučení informuje uživatele o nabídkách a produktech, o které by uživatel mohl mít zájem. Může být aplikováno v podobě zvýraznění textu nebo přidáním části obsahu s doporučeními.

Náhrada obsahu může být přínosem ve chvíli, kdy uživatel nemůže nebo nechce pracovat s obsahem, který na stránce je. [10] zmiňuje automatické generování alternativního textu na základě potřeb uživatele (například jeho úrovně znalostí). Využitelnost takové metody v obecném frameworku je v současnosti technicky složitá a nebude součástí této práce. Varianty zobrazení obsahu je možné použít, pokud tvůrce webu připravil pro nějakou stránku nebo objekt různé varianty textu.

Zvýraznění částí textu je v podstatě forma doporučení uživateli, které části zobrazených informací by mohly být pro uživatele zajímavé.

#### Adaptace způsobu prezentace

Změna způsobu prezentace může být například změna velikosti písma, zvýšení kontrastu písma, či změna formátu stránek. Změnou způsobu prezentace je také nahrazení textových informací audiem.

#### Adaptace struktury

Řazení odkazů. Pokud je framework schopen zjistit, jaké odkazy vyskytující se na stránce by uživatel mohl preferovat, může je seřadit podle relevance. U této adaptace je třeba počítat s tím, že ne všechny odkazy můžou být řazeny. Odkazy, které jsou součástí z textu, nelze libovolně vytrhnout z kontextu.

Stejně jako řazení odkazů je i popis odkazů metoda pro rozlišování odkazů na základě jejich relevance. Ke každému odkazu může být přidán jeho popis (příp. Ikona), který určí jeho relevanci pro uživatele.

Skrývání/zablokování/odstranění odkazů se také týká relevance odkazů. Odkazy, které budou vyhodnoceny jako nerelevantní, budou odstraněny, zablokovány nebo skryty. Odstraněním odkazu se rozumí jeho úplné odstranění včetně textu. Zablokovaný odkaz na stránce zůstane, ale nebude možné ho použít. Skrytý odkaz zůstane funkční, ale není barevně ani jinak odlišen od ostatního textu.

Generování odkazů může být použito na stránce, kde framework rozhodne o tom, že by zde mělo existovat propojení a tvůrce webové aplikace zde odkaz neudělal.

Přímá navigace je metoda, která provází uživatele po webu a slouží jako průvodce. Na každé navštívené stránce je tlačítko "vpřed", po použití tlačítka bude uživatel přesměrován na další stránku.

## 2.3 Rozhodování o provádění adaptací

Většina existujících systémů pro adaptování webových aplikací pracuje online s aktuálním návštěvníkem webové aplikace. Rozhodování o provádění adaptací je tedy součástí inline mining. Online práce spočívá ve třech úkolech:

1. pozorování uživatele, získání záznamu o jeho chování
2. klasifikace uživatele
3. určení vhodných adaptací

#### Získání záznamu o chování uživatele

Pozorování uživatele obvykle probíhá stejným způsobem jako získávání dat pro web usage mining, tím se získá model aktuálního uživatele.

#### Klasifikace uživatele

Smyslem klasifikace je najít vhodný vzorový uživatelský model nebo modely z těch vytvořených pomocí data mining, které proběhlo offline (obr. 2.1).

#### Určení vhodných adaptací

V nalezených vzorových modelech jsou objeveny stránky či objekty, které aktuální uživatel dosud nepoužil a ty se stanou předmětem adaptací a budou adaptacemi propagovány.

# 3 Použité metody

Při rozhodování o úpravě webových aplikací framework používá jako hlavní zdroj data o tom, jak je webová aplikace používána, tedy web usage mining (viz 2.1.1). Ze tří typů web mining (web usage mining, web content mining, web structure mining) v současnosti jedině web usage mining dokáže poskytnout celkový pohled na životní cyklus webové aplikace, a proto je vhodný jako zdroj pro relevantní adaptace. Pokud bychom si dali za cíl adaptovat webovou aplikaci na základě web content mining a web structure mining, bylo by potřeba strojově pochopit textové informace uvnitř stránek webové aplikace a jejich spojitost s ostatními stránkami. Tento postup daleko je za hranicemi dnešních technologií. Pokus o provádění adaptací na základě web content mining a web structure mining není příliš perspektivní. Framework by měnil webovou aplikaci, aniž by chápal její podstatu a nebyl by schopen tento hendikep dostatečně vyvážit. Tím by se snažil být chytřejší než tvůrce webové aplikace, což je v tomto případě těžko opodstatněné. Naopak díky web usage mining má framework k dispozici informace o používání webové aplikace řadou uživatelů, z nichž může těžit hodnotné informace, které tvůrce nezná.

Metody web content mining a web structure mining nejsou použity, jejich přínos jako doplněk web usage mining by mohl být platný, implementace by však byla poměrně komplikovaná a je nad rámec této práce. Web content mining je částečně nahrazeno použitím vlastností stránek (viz 3.5).

Snaha kategorizovat uživatele podle jejich společných rysů neboli podobností vyskytujících se v jejich transakčních historiích je základní myšlenkou činnosti frameworku. Pro hledání byly použity metody hledání asociačních pravidel a sekvenčních vzorů a kolaborativní filtrování.

## 3.1 Asociační pravidla

Hledání asociačních pravidel je v oboru dobývání znalostí jednou z nejznámějších metod pro objevování zajímavých vztahů v transakčních databázích. Definice asociačních pravidel [16] je následující: Nechť

$$I = \{i_1, i_2, ..., i_m\}$$

je množina všech položek tedy webových stránek aplikace.

$$T = \{t_1, t_2, ..., t_n\}$$

je množina transakcí, kde každá transakce $t_i$ je taková množina webových stránek, že $t_i \subseteq I$. Asociační pravidlo je definováno jako

$$X \Rightarrow Y, \text{ kde } X \subset I, Y \subset I, \text{ a zároveň } X \cap Y = \emptyset.$$

*Podpora* asociačního pravidla $X \Rightarrow Y$ je procento transakcí v $T$, které obsahují $X \cup Y$. Počet transakcí v $T$, které obsahují $X$ je značen $Xpočet$. Pokud $n$ je počet transakcí v T, pak se *podpora* pravidla $X \Rightarrow Y$ spočítá takto:

$$podpora(X \Rightarrow Y) = \frac{(X \cup Y)počet}{n}.$$

*Spolehlivost* asociačního pravidla $X \Rightarrow Y$ je procento transakcí v $T$ obsahujících $X$, které obsahují i $Y$. Spočítá se následujícím způsobem:

$$spolehlivost(X \Rightarrow Y) = \frac{(X \cup Y)počet}{Xpočet}.$$

Za vyhledávání asociačních pravidel(association rules) z transakčního logu webové aplikace je zodpovědný samostatný modul využívající algoritmus *Apriori* [26]. Vstupem modulu je kromě samotného transakčního logu také hodnota udávající minimální podporu a hodnota udávající minimální spolehlivost. Výstupem modulu je soubor s asociačními pravidly, kde každé z nich má podporu resp. spolehlivost vyšší nebo rovnou minimální hodnotě podpory resp. spolehlivosti. Hodnoty minimální podpory a minimální spolehlivosti pro asociační pravidla jsou parametricky nastavitelné, neboť různé aplikace mohou mít diametrálně odlišné potřeby či využití a prahy nastavené pro jednu aplikaci nemusí být vhodné pro jinou aplikaci, tato volba je ponechána tvůrci webové aplikace.

Modul pro hledání asociačních pravidel, je spouštěn během offline mining. Výsledná asociační pravidla jsou pak používána při online interakci s jednotlivými uživateli. Při provozu zkušební webové aplikace online porovnávání transakcí uživatelů s asociačními pravidly probíhá natolik rychlým způsobem, že plynulost webové

aplikace není ovlivněna a odezva frameworku je dostatečně rychlá.

Kromě rychlosti při provozu je další výhodou asociačních pravidel jejich možná škálovatelnost pomocí nastavování různých hodnot minimální podpory a spolehlivosti. Vzhledem k tomu, že povaha webové aplikace není dopředu známa, je tato vlastnost velmi podstatná.

## 3.2 Sekvenční vzory

Sekvenční vzory (sequential patterns) je další metodou dobývání znalostí z transakčních databázích. Je podobná vyhledávání asociačních pravidel, narozdíl od nich ovšem zohledňuje pořadí položek v transakcích. U definice sekvenčních pravidel [9] použijeme stejnou množinu položek $I$ jako u asociačních pravidel:

$$I = \{i_1, i_2, ..., i_m\}$$

dále

$$T =< p_1, p_2, ..., p_n >$$

je množina transakcí, kde každá transakce $p_i$ je posloupnost webových stránek $< p_{i1}, p_{i2}, ..., p_{il} >$ a pro každé $1 \leq j < k \leq l$ platí že uživatel v rámci transakce T navštívil stránku $p_{ij}$ dříve než stránku $p_{ik}$. Sekvenčním pravidlem se pak rozumí: $X \Rightarrow y$, kde $X$ je posloupnost prvků $< x_1, x_2, ..., x_s >$, taková že pro $1 \leq i \leq s$ platí

$$x_i \in I \text{ a } y \subset I.$$

Pro měření kvality sekvenčního vzoru budeme používat *podporu sekvence* a *spolehlivost sekvenčního vzoru*. Počet transakcí v $T$, které obsahují sekvenci $X$, je značen $Xpočet$. *Podpora sekvence* se spočítá:

$$podpora(X) = \frac{(X)počet}{n}.$$

Označme $Aq$ posloupnost, která vznikla napojením prvku $q \in I$ na posloupnost A složenou z $s$ prvků ($A =< x_1, x_2, ..., x_s >$). Aq je pak tedy posloupnost $< x_1, x_2, ..., x_s, q >$. *Spolehlivost sekvenčního vzoru* $X \Rightarrow y$ se spočítá následujícím způsobem:

$$spolehlivost(X \Rightarrow y) = \frac{(Xy)počet}{Xpočet}.$$

O vyhledávání sekvenčních vzorů se stejně jako u asociačních pravidel stará samostatný modul, který používá algoritmus *GSP* [9]. Algoritmus *GSP* je modifikace algoritmu *Apriori*, vstup, výstup i používání jsou tedy velmi podobné. I u sekvenčních vzorů se nastavují hodnoty minimální podpory a minimální spolehlivosti. Algoritmus *GSP* se spouští offline, výsledkem je soubor vzorů, které se používají při online provozu. Pro plynulost provozu frameworku platí to samé jako u asociačních pravidel.

Vyhledávání sekvenčních vzorů je metoda rychlá při online používání, tudíž vhodná pro použití ve frameworku. Její výsledky můžou být použity jako doplněk k asociačním pravidlům nebo samostatně. Volba je ponechána na tvůrci webové aplikace, u některých webových aplikací může být informace o chronologii transakcí důležitá, zatímco u jiných ne.

## 3.3 Kolaborativní filtrování

Kolaborativní filtrování [9] je metoda, která na základě transakcí aktuálního uživatele prohledá transakční historie předešlých uživatelů a snaží se najít transakční historie podobné transakční historii aktuálního uživatele. Z takto nalezených podobných transakcí vzejde doporučení aktuálnímu uživateli, ve němž budou ty transakce z podobných transakčních historií, které aktuální uživatel dosud nepoužil. Tento typ kolaborativního filtrování se nazývá *user-based*, dosahuje dobrých výsledků, ale pro aplikace s velkým množstvím uživatelů je časově náročná [23]. Proto je tato metoda pro používání online, kdy výsledky kolaborativního filtrování jsou potřebné v reálném čase, nevhodná.

Vhodnější typ kolaborativního filtrování je *item-based*. Nehledá skupinu podobných uživatelů, ale skupinu podobných transakcí podle toho, které transakce byly provedeny spolu v rámci jedné transakční historie. Použitý algoritmus je *k-nearest neighbor algoritmus*, podobnost dvou položek je definována takto:

$$s(i,j) = \frac{\vec{i} \cdot \vec{j}}{|\vec{i} \cdot \vec{j}|},$$

kde $i \in I, j \in I$ a $\vec{x} = (x_1, x_2, ..., x_n)$ a pro $1 \leq k \leq n$ platí, že $x_k = 0$ pokud $x$ není v transakci $t_k$, jinak $x_k = 1$.

Podle [23] [24] jsou výsledky *item-based* kolaborativní filtrování minimálně stejně kvalitní jako *user-based* a navíc nároky na výkon jsou natolik nižší, že by měly umožňovat fungování v reálném čase.

## 3.4 Ostatní metody

Žádná z dalších metod zmíněných v úvodu nebyla použita. Co se týče klasifikace uživatelů (clustering and classification), tak tato metoda nabízí velmi podobnou funkčnost jako kolaborativní filtrování. Navíc má oproti kolaborativnímu filtrování vysoké nároky na výkon systému [23]. Především z těchto dvou důvodů nebyla klasifikace uživatelů implementována.

Užitečnost metody path analysis je poměrně sporná. Sice je možné, že se ve webové aplikaci určité cesty procházení budou objevovat opakovaně, nicméně provádět odhady a predikce, která cesta bude použita, jsou komplikované a vzhledem k předpokladané malé četnosti opakování cest i málo spolehlivé. Výsledky, jaké by metoda path analysis poskytla, jsou navíc dostupné pomocí sekvenčních vzorů.

## 3.5 Transkační historie

U většiny pokročilých systému pro adaptování webových aplikací jsou zdrojem informací o chování uživatele navštívené stránky. Framework jde dál a zaznamenává i akce prováděné uživatelem na samotné stránce. Příkladem takové akce je třeba dynamické rozbalení původně skrytého obsahu. Důsledkem takového přístupu je transakční historie, která je jako celek tvořena transakcemi různého typu. Zacházení s ní jako s celkem může při web usage mining nést svá rizika, transakce jednoho typu mohou vypovídat o jiných věcech než akce jiného typu. Například navštívené stránky indikují o jaký typ informací má uživatel zájem, zatímco objekty, které uživatel na stránce používá, mohou vypovídat o tom, v jaké formě chce uživatel informace dostávat. Zjistit tedy, o jaký typ informací má uživatel zájem, bude jednoduší z transakční historie, v níž budou jen navštívené stránky. Z tohoto důvodu metody web usage mining ve frameworku nečerpají pouze z celkové transakční historie, ale

i z částečných transakčních historií, tedy historií obsahujících pouze jednotlivé typy transakcí.

Zvláštním typem transakce je pak načtení stránky. Stránky totiž mohou mít definované vlastnosti. K tomu je použit zvláštní metatag umístěný tvůrcem webové aplikace do hlavičky stránky. Z uživatelem navštívených stránek tak může být vytvořena speciální transakční historie vlastností, kde se transakcí myslí vlastnost navštívené stránky.

Tři metody web usage mining implementované ve frameworku poskytují předpověď transakcí, které by mohl uživatel v budoucnu provést. U asociačních pravidel a sekvenčních vzorů je navíc k dispozici číslo uvádějící míru spolehlivosti každé předpovědi. Použití různých metod může mít různé výsledky. Obecně není možné říci, která z metod bude vhodná pro určité typy částečných transakčních historií, transakční historii vlastností či celkovou transakční historii. Volba metod je ponechána na tvůrci webové aplikace stejně jako váha, kterou se rozhodne přikládat výsledkům jednotlivých metod.

## 3.6 Adaptace

Výsledkem práce frameworku jsou tedy predikce či odhady transakcí, které uživatel v budoucnu provede. Tyto odhady framework nijak neinterpretuje, součástí činnosti frameworku není provádění adaptací. Odhady transakcí framework předá webové aplikaci, kde zůstává na jejím tvůrci, jak s výsledky naloží a jaké adaptace použije.

Důvodem pro nezahrnutí adaptací do frameworku je fakt, že dopředu není známo nic o samotné webové aplikaci. Bez znalosti účelu webové aplikace, cílové skupiny, požadavků provozovatele webové aplikace či jejích omezení, by rozhodnutí provést jakoukoliv adaptaci (viz 2.2.1) bylo neuvážené.

# 4 Architektura frameworku

Sledování akcí uživatele, ukládání akcí uživatele a analyzování akcí uživatele jsou tři hlavní činnosti, které framework vykonává. Interpretace analýzy akcí uživatele je ponechána na tvůrci webové aplikace.

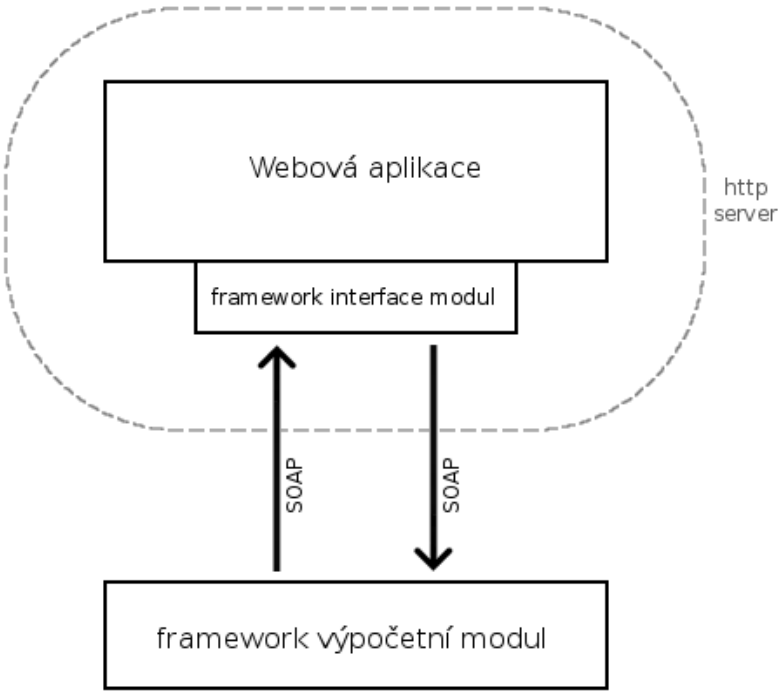
Jedním z cílů je, aby framework bylo možné využívat pouze s pomocí technologií dostupných na běžném webhostingu. Implementace metod z kapitoly 3 by ovšem těmito technologiemi nebyla příliš efektivní a patrně by znamenala pro server webové aplikace příliš vysokou zátěž. I z těchto důvodů byla při návrhu frameworku vybrána architektura typu klient-server, kde kde část frameworku starající se o data mining je fyzicky oddělena od webové aplikace.

Framework je složen ze dvou částí:

**Interface modul** zastává v architektuře klient-server roli klienta. Poskytuje javacsriptové rozhraní, které se stará o sledování akcí uživatele a zároveň dává tvůrci webové aplikace přístup k výsledkům výpočetního modulu frameworku.

**Výpočetní modul** funguje jako server a stará se o analýzu chování uživatelů na základě přístupového logu z webové aplikace. Na přístupový log aplikuje metody web usage mining, jejichž výsledky následně v reálném čase aplikuje na záznamy o uživatelích, kteří aktuálně používají webovou aplikaci.

Oba moduly jsou od sebe fyzicky odděleny a nemusí se nacházet na jednom serveru. Na serveru, kde je umístěna webová aplikace je umístěn *interface modul*, ten s *výpočetním modulem* komunikuje pomocí protokolu SOAP [22], jak je znázorněno na obrázku 4.1. Tato architektura umožňuje snadné nasazení frameworku na jakoukoliv webovou aplikaci, což je jedna z priorit. *Interface modul* používá technologie, které jsou dnes běžně dostupné u poskytovatelů webhostingu. Složitější nástroje jsou pak použity ve *výpočetním modulu*, který funguje jako služba a pro tvůrce webové aplikace tudíž odpadá nutnost složité instalace. Server, na němž je *výpočetní modul*, pak nemusí být spojen pouze s jednou webovou aplikací, teoreticky může poskytovat výpočetní kapacitu pro více webových aplikací zároveň.




Obr. 4.1: Architektura frameworku

## 4.1 Interface modul

Pomocí *interface modulu* má tvůrce webové aplikace možnost určit si jaké transakce budou sledovány, případně si definovat vlastní transakce a také může získat výsledky analýzy z výpočetního modulu. Tvůrce webu přijde do styku pouze s *interface modulem*, o existenci *výpočetního modulu* v podstatě nemusí vůbec vědět. Má přístup pouze k funkcím *interface modulu* a co se děje za ním se ho netýká.

*Interface modul* je napsaný převážně v javascriptu, malá část kódu je v PHP. Modul musí inicializovat tvůrce webu ve zdrojovém kódu každé stránky. *Interface modul* zajišťuje sledování transakcí, které uživatelé provádějí. K tomu používá metody DHTML (viz kapitola 2.1.1), každá zaznamenaná transakce je předána *výpočetnímu modulu*, kde je udržován transakční log. K identifikaci jednotlivých návštěvníků se používají cookies. Tím, že pro sledování transakcí se používá DHTML, odpadají problémy s možným cachováním stránek (viz kapitola 2.1.1).

Pro zpracovávání výsledků přijatých z *výpočetního modulu* je nutné, aby si tvůrce webové aplikace vytvořil javascriptovou funkci, která výsledky interpretuje a

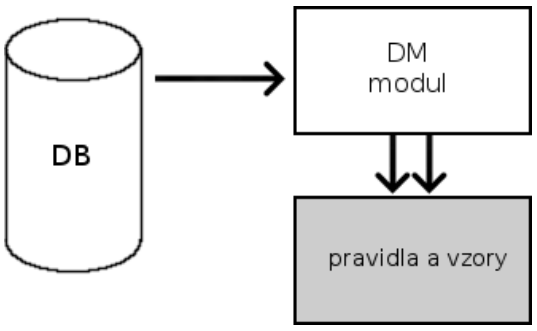
případně provede adaptace webové aplikace. Výsledky obsahují pole doporučených transakcí a vah, které určují kvalitu jednotlivých doporučení. Toto pole je předáno jako paramatetr funkci, kterou si tvůrce webu pro zpracování vytvořil.

## 4.2 Výpočetní modul modul

*Výpočetní modul* je klíčovou částí celého frameworku. Jeho úkolem je dobývání znalostí ze zaznamenaných transakcí uživatelů a také jejich aplikace na konkrétní uživatele. Vykonává tedy online i offline mining (viz kapitola 2.1.1).

*Výpočetní modul* není jednolitý kus kódu, je to logické označení pro skupinu komponent, v níž každá komponenta plní určitou úlohu a které používají společnou databázi. Všechny části *výpočetního modulu* jsou implementovány pod operačním systémem Linux. *Výpočetní modul* obsahuje tyto části:

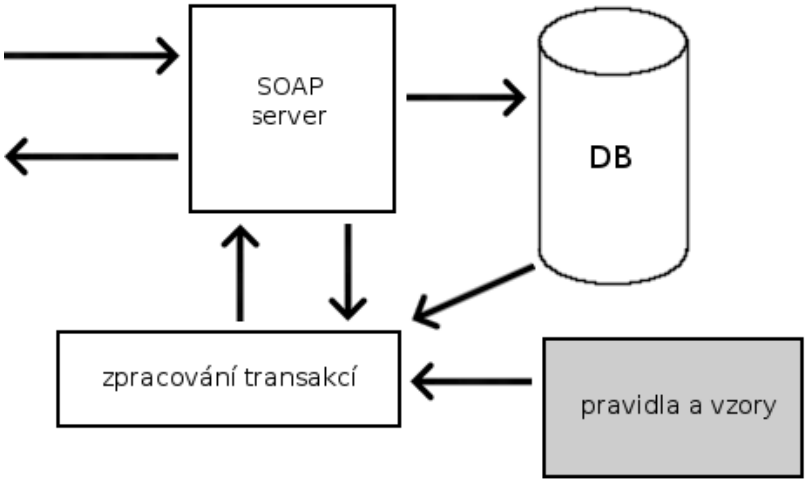
- databáze uživatelských transakcí
- SOAP server
- data mining modul (DM modul)
- modul pro analýzu transakcí




Obr. 4.2: Dobývání znalostí z databáze uživatelských transakcí

Klíčovou funkcí *výpočetního modulu* je dobývání znalostí z přístupového logu webové aplikace, který je uložen v databázi. Vstupem tohoto procesu je záznam uživatelských transakcí a výstupem je soubor zajímavých vzorů a pravidelností objevujících se v transakcích. Schéma práce je znázorněno na obrázku 4.2. Podmínkou pro provádění dobývání znalostí je dostatečně naplněná databáze uživatelských transakcí.

Proběhnutí procesu dobývání znalostí je prerekvizitou pro analýzu transakcí, o analýzu se stará *modul pro analýzu transakcí*. Ta se již zabývá konkrétními transakcemi v reálném čase, které právě probíhají ve webové aplikaci a uplatňuje na ně poznatky získané při dobývání znalostí. Které metody z kapitoly 3 se použijí, je nastaveno v konfiguračním souboru. Práce *výpočetního modulu* je ukázána na obrázku 4.3 Nejprve SOAP server (více o SOAP v kapitole 4.3) přijme z webové aplikace informaci o tom, jakou transakci uživatel právě udělal a ta je následně uložena do databáze. Pokud spolu s informací o provedené transakci byla z webové aplikace poslána žádost o provedení analýzy. SOAP server spustí *modul pro analýzu transakcí*. Ten z databáze vyzvedne kompletní transakční historii aktuálního uživatele. Poté v závislosti na nastavení konfiguračního souboru se pro aktuální transakční historii hledají vyhovující asociační pravidla, sekvenční vzory nebo se provede kolaborativní filtrování. Výsledkem je sada doporučených transakcí. V případě, že byly použity asociační pravidla nebo sekvenční vzory, je k nim k dispozici údaj o spolehlivosti pravidla (vzoru). Doporučené transakce jsou předány SOAP serveru, který je odesílá webové aplikaci. Jak konkrétně spolupracují jednotlivé komponenty při analýze transakcí je popsáno v kapitole 5.2.4.




Obr. 4.3: Analýza transakcí

## 4.3 Komunikace *interface modulu* s *výpočetním modulem*

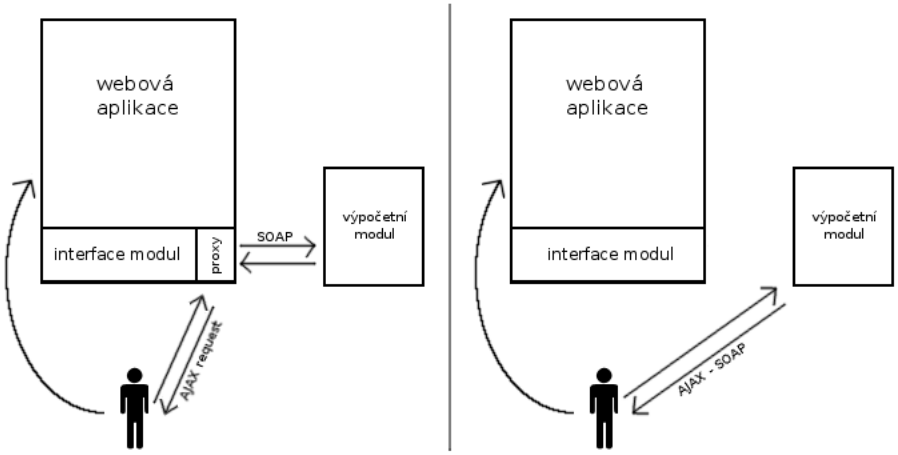
Jak již bylo zmíněno, *výpočetní modul* je od *interface modulu* oddělen fyzicky, tedy může se nacházet na jiném počítači. Vzniká tudíž potřeba vyřešit způsob komunikace mezi oběma moduly. Komunikace by měla splňovat následující kritéria:

- na straně *interface modulu* smějí být použity technologie dostupné na běžném webhostingu
- dostatečně rychlá doba odezvy
- možnost předávat libovolná textová data oběma směry

SOAP (Simple Object Access Protocol)[22] je technologie, která tyto požadavky splňuje. Pro SOAP komunikaci je použit protokol HTTP a šablona, kde *interface modul* je klient a *výpočtení modul* je server. SOAP používá pro přenos zpráv formát XML, do zpráv se tedy dají vložit libovolné textové informace.

Komunikace pomocí technologie je natolik rychlá, aby plynulost používání webové aplikace nebyla narušena. V ukázkové webové aplikaci doba od odeslání požadavku do přijetí odpovědi nepřesáhla jednu vteřinu.

SOAP požadavky je možné posílat prostřednictvím javascriptu a technologie AJAX[17], což ovšem nefunguje, pokud se *výpočtení modul* nachází na jiné doméně než *interface modul* kvůli bezpečnostním opatřením prohlížečů. Prohlížečům nedovoluje používat AJAX napříč doménami pravidlo "Same origin policy"[17][19]. Ačkoliv se do budoucna uvažuje o zrušení tohoto opatření[18], v současnosti je nutné toto omezení obejít. *Interface modul* tedy používá PHP proxy, AJAXem je zavolán PHP skript, který je umístěn na stejném serveru jako webová aplikace a ten obstará SOAP komunikaci(obr. 4.4 vlevo). Pokud bude v budoucnu pravidlo "Same origin policy" zrušeno, bude možné krok volání PHP skriptu vynechat(obr. 4.4 vpravo), vznikne tak ovšem potenciální bezpečnostní riziko, kterému bude nutné věnovat pozornost.




Obr. 4.4: Schéma komunikace s *výpočtením modulem*. Vpravo s použitím proxy. Vlevo bez použití proxy.

# 5 Implementace

V této kapitole jsou detailněji popsány *interface modul* a *výpočetní modul*.

## 5.1 Funkce interface modulu

Jak již bylo řečeno, *interface modul* je část frameworku, ke které může tvůrce webové aplikace přímo přistupovat. Zdrojové kódy *interface modulu* jsou v adresáři `sp_pack/`. Adresář `sp_pack/` se nachází v kořeni adresářové struktury webové aplikace. Každá stránka webové aplikace musí mít v hlavičce inicializační skript, který umožní fungování frameworku.

```
<script type="text/javascript"
        src="app_url/sp_pack/sp.js"></script>
<script type="text/javascript">
  sp.init({
    show_control : false
  });
</script>
```

Javascriptová třída `sp` reprezentuje *interface modul*, její metoda `init()` provádí inicializaci modulu. Pomocí parametrů metody `init()` lze nastavit chování frameworku. Parametry jsou ve formátu JSON [20]. Metoda může mít tyto parametry:

- `show_control` - příznak, zda zobrazovat v browseru zobrazovat informace o práci frameworku
- `log_settings` - nastavení, které typy událostí má automaticky framework zaznamenávat, resp. odesílat ke zpracování *výpočetnímu modulu*, možné hodnoty jsou:
  - `loads` - zaznamenává se pouze načtení jednotlivých stránek v prohlížeči, defaultně nastavená hodnota
  - `loads_ajax` - zaznamenává se načtení jednotlivých stránek v prohlížeči a načtení stránek pomocí AJAXu

`loads_objects` - zaznamenává se načtení jednotlivých stránek a manipulace s objekty na stránce

`all` - zaznamenává se vše

Kromě zmíněných tří typů událostí je umožněno zaznamenávat ještě události definované uživatelem(viz metoda `log_ud_event()`).

- `compute_loads` - přepínač pro zapnutí/vypnutí analýzy načtených stránek, přípustné hodnoty jsou `true/false`, defaultní hodnota je `true`
- `compute_ajax` - přepínač pro zapnutí/vypnutí analýzy použití AJAXu, přípustné hodnoty jsou `true/false`, defaultní hodnota je `false`
- `compute_obj` - přepínač pro zapnutí/vypnutí analýzy manipulace s objekty na stránkách, přípustné hodnoty jsou `true/false`, defaultní hodnota je `false`
- `compute_ud` - přepínač pro zapnutí/vypnutí analýzy uživatelsky definovaných událostí, přípustné hodnoty jsou `true/false`, defaultní hodnota je `false`
- `compute_properties` - přepínač pro zapnutí/vypnutí analýzy vlastností navštívených stránek, přípustné hodnoty jsou `true/false`, defaultní hodnota je `false` (více o vlastnostech navštívených stránek v kapitole Vlastnosti stránek)
- `compute_all` - přepínač pro zapnutí/vypnutí analýzy veškerých provedených transakcí dohromady, přípustné hodnoty jsou `true/false`, defaultní hodnota je `false`
- `adapt_loads_handler` - určí funkci, která bude zpracovávat a aplikovat výsledky *výpočetního modulu* obsahující doporučení akcí načtení stránek
- `adapt_ajax_handler` - určí funkci, která bude zpracovávat a aplikovat výsledky *výpočetního modulu* obsahující doporučení použití AJAXu
- `adapt_obj_handler` - určí funkci, která bude zpracovávat a aplikovat výsledky *výpočetního modulu* obsahující doporučení manipulace s objekty
- `adapt_ud_handler` - určí funkci, která bude zpracovávat a aplikovat výsledky *výpočetního modulu* obsahující doporučení uživatelsky definovaných událostí
- `adapt_properties_handler` - určí funkci, která bude zpracovávat a aplikovat výsledky *výpočetního modulu* obsahující doporučení ohledně vlastností stránek
- `adapt_all_handler` - určí funkci, která bude zpracovávat a aplikovat výsledky *výpočetního modulu* obsahující doporučení veškerých možných transakcí

- `disable_jquery_override` - může mít hodnoty `true/false`, povoluje či zakazuje úpravu jQuery metod, pokud je hodnota `false`, tak není možné zaznamenávat manipulace s objekty a načítání stránek pomocí AJAXu, defaultní hodnota je `true`
- `results_ondemand_only` - může mít hodnoty `true/false`, určuje kdy se získají výsledky analýzy chování uživatele, pokud je `true`, musí se zavolat metoda `get_results`, pokud je `false` výsledky se získají vždy jako odpověď *výpočetního modulu* na záznam každé události, defaultní hodnota je `false`

### 5.1.1 Přehled dalších metod API frameworku

`log_ud_event(name, pars)` - zaznamenává uživatelsky definované události, argument `name` udává název události, argument `pars` je možné použít pro další parametry

`get_user_results()` - odešle *výpočetnímu modulu* požadavek na získání analýzy chování aktuálního uživatele; analýza zpracovává celý záznam transakční historie uživatele

`get_user_results_ctrans(transaction)` - odešle *výpočetnímu modulu* požadavek na získání analýzy chování aktuálního uživatele, analýza zpracovává celý záznam transakční historie uživatele, ovšem ve výsledných pravidlech musí být transakce zadaná parametrem `transaction`

### 5.1.2 Sledování akcí uživatelů a vnitřní práce *interface modulu*

Jako většina systémů pro adaptaci webových aplikací i framework pro pozorování aktivity uživatelů vychází z logu načtených stránek. Vzhledem k tomu, že v současnosti je velmi rozšířené používání DHTML technologií, snaží se framework sledovat i jiné události související s konáním uživatele. Fungování frameworku je velmi úzce spojeno javascriptovou knihovnou jQuery [21], a to nejen tím, že využívá funkce jQuery, ale i tím, že umožňuje sledovat používání funkcí jQuery na straně webové aplikace. Díky tomu je framework schopen získat ucelenější a podrobnější přehled o používání webové aplikace. Pokud není při inicializaci frameworku zakázáno logování funkcí jQuery (přepínač `disable_jquery_override`), pak framework zaznamenává používání některých jQuery funkcí v závislosti na nastavení

parametru `log_settings` při inicializaci. Jedná se o funkce pro načítání stránek AJAXem a dále o funkce, které manipulují s objekty na webové stránce (např. `show()`, `hide()`, `toggle()`, atd.). Zaznamenávání těchto funkcí poskytuje lepší popis chování uživatelů webových aplikací, které méně či více používají DHTML prvky.

### 5.1.3 Vlastnosti stránek

Další možnost pro tvůrce webu pro tvůrce webové aplikace. jak nastavit *výpočetní modul*, je použití vlastností stránek. Vlastnosti se definují v hlavičce pomocí speciálních metatagů. Každé stránce v rámci webové aplikace lze nastavit libovolný počet vlastností. Formát pro metatagy vlastností je `<meta name="sp_value.nazev_vlastnosti" content="hodnota" />`, příklad použití vlastností může vypadat takto:

```
<meta name="sp_value.type" content="product" />
<meta name="sp_value.category" content="1972" />
<meta name="sp_value.price" content="2" />
```

Použitý příklad pochází z ukázkové aplikace (eshop viz kapitola 6). Vlastnost page udává, jakého typu stránka je, hodnota `product` znamená, že jsme v detailu produktu. Vlastnost `category` říká, ve které kategorii se produkt nachází. A nakonec vlastnost `price` určuje cenové zařazení produktu v rámci kategorie na stupnici od 0 do 10. Nejlevnější produkt v kategorii má hodnotu `price` 0, nejdražší naopak 10.

Vlastnosti stránek jsou ve *výpočetním modulu* podrobeny stejnému procesu dobývání znalostí jako ostatní data. Z pohybu uživatele po stránkách webové aplikace je pro každou vlastnost vygenerována transakční historie. V příkladu eshopu to bude například pro vlastnost `category` záznam kategorií, kterými uživatel prošel.

Vzhledem k faktu, že vlastnosti jsou použity jako transakce, je vhodné, aby hodnoty vlastností byly diskrétní. Z toho důvodu je v příkladu pro vlastnost `price` použita stupnice 0 až 10 na místo skutečné ceny.

### 5.1.4 Některé další vnitřní funkce *interface modulu*

`save_log(action)` - funkce pro uložení akce uživatele, pro popis akce je k dispozici argument `action`, který musí být ve formátu JSON; informace o akci jsou protokolem SOAP (viz kapitola Komunikace s *výpočetním modulem*) odeslány *výpočetnímu modulu*, kde jsou uloženy; pokud má argumentu `results_ondemand_only` metody `init()` hodnotu `false`, pak *výpočetní modulu* odešle zpět v odpovědi na SOAP požadavek výsledky analýzy chování uživatele a tento výsledek je zpracován funkcí `aply_adaptations()`
`aply_adaptations()` - funkce je volána poté, co *výpočetní modul* vrátí výsledky, ty jsou v této funkci rozděleny podle typu (načtení stránek, AJAX, manipulace s objekty, uživatelsky definované, vlastnosti) a na jednotlivé části je zavolána přislušná uživatelská funkce definovaná v metodě `init()` parametry `adapt_loads_handler`, `adapt_ajax_handler`, `adapt_obj_handler`, `adapt_ud_handler` a `adapt_properties_handler`

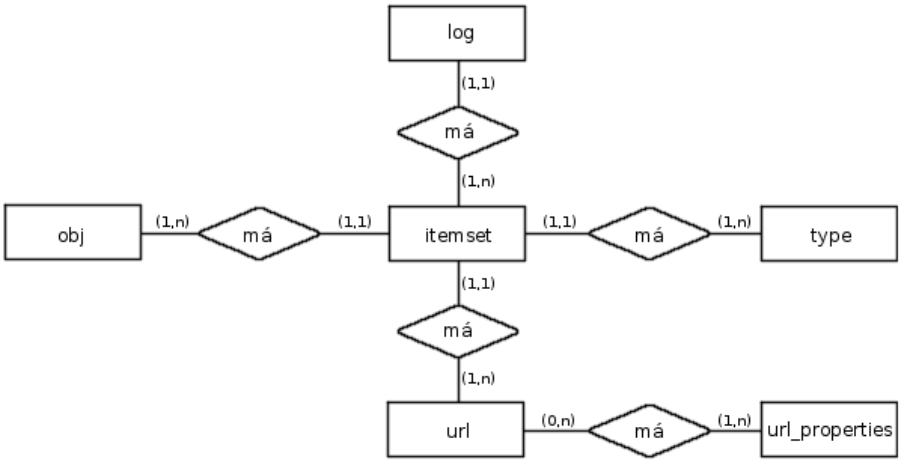
## 5.2 Funkce výpočetního modulu

Jak bylo zmíněno *výpočetní modulu* má čtyři části:

- databáze uživatelských transakcí
- SOAP server
- data mining modul (DM modul)
- modul pro analýzu transakcí

### 5.2.1 Databáze uživatelských transakcí

Databáze slouží nejenom k ukládání transakcí uživatelů webové aplikace, ale plní také pomocnou funkci při formátování transakcí pro metody použité při dobývání znalostí v *DM modulu*. Položky v transakčním logu získané z webové aplikace jsou většinou textové, použité implementace algoritmů pro dobývání znalostí (např. Apriori) potřebují pracovat s transakcemi, které jsou identifikovány celými čísly. Ve frameworku je použit databázový systém MySQL.
**Tabulka log** - slouží jako transakční log, ukládají se sem veškeré transakce uživatelů v textové podobě. Jsou v ní následující položky:




Obr. 5.1: E-R model databáze

uid - identifikátor uživatele

ts - čas akce

type - typ transakce uživatele (např. load - načtení stránky,ajax - použití AJAXu)

url - adresa stránky, kde se akce stala

obj - doplňující informace o akci (např. identifikace objektu se kterým se manipuluje nebo parametry při použití AJAXu)

user - říká, zda se jedná o uživatelsky definovanou událost, může nabývat hodnot 0 nebo 1

**Tabulka itemset** - záznamy v této tabulce odpovídají transakcím resp. akcím uživatelů, které se ve webové aplikaci mohou objevit. Obsahuje tyto položky:

id - celočíselný identifikátor transakce

type - reference na typ transakce do tabulky *sp_type*, celé číslo

url - reference na url transakce do tabulky *sp_url*, celé číslo

obj - reference na doplňující informace o transakci do tabulky *sp_obj*, celé číslo

**Tabulka sp_type** - pomocná tabulka k tabulce *itemset*

id - celočíselný identifikátor typu uživatelské transakce, odpovídá položce *type* v tabulce *itemset*

type - textová hodnota, název typu uživatelské transakce, obsahuje stejné hodnoty jako položka *type* v tabulce *log*

**Tabulka sp_url** - pomocná tabulka k tabulce *itemset*

id - celočíselný identifikátor url uživatelské transakce, odpovídá položce *url* v tabulce *itemset*

url - textová hodnota, url uživatelské transakce, obsahuje stejné hodnoty jako položka *url* v tabulce *log*

title - název stránky, odpovídá tagu `<title>`

**Tabulka sp_obj** - pomocná tabulka k tabulce *itemset*

id - celočíselný identifikátor pro doplňující informace uživatelské transakce, odpovídá položce *obj* v tabulce *itemset*

obj - textová hodnota, doplňující informace uživatelské transakce, obsahuje stejné hodnoty jako položka *obj* v tabulce *log*

Přítomnost tabulek *sp_type*, *sp_obj* a *sp_url* je na první pohled zbytečná. Logické řešení by bylo v tabulce *itemset* použít pro položky *type*, *obj* a *url* textové hodnoty namísto celočíselných. Pokud by tak bylo, došlo by k omezení rychlosti některých dotazů do databáze. Často používaný dotaz `SELECT id FROM itemset WHERE type='text_type' AND url='text_url' AND obj='text_obj'` pro optimální rychlost v MySQL potřebuje mít vytvořený index nad sloupci *type*, *obj* a *url* zároveň, což pro textové sloupce v MySQL není možné, ale pro celočíselné hodnoty to problém není.

**Tabulka url_properties** - obsahuje záznamy hodnot vlastností stránek

id - identifikátor

property - název vlastnosti stránky

value - hodnota vlastnosti stránky

**Tabulka sp_url_url_properties** - propojení tabulek *sp_url* a *url_properties*

id - identifikátor

url - reference do tabulky *sp_url* na položku *id*

property - reference do tabulky *url_properties* na položku *id*

### 5.2.2 Data mining modul

Hlavní činností *data mining modulu* je hledání asociačních pravidel, sekvenčních vzorů, případně příprava dat pro kolaborativní filtrování.

### Data mining

Pro hledání asociačních je použit program nazvaný `apriori`, který jak je z názvu patrné provádí algoritmus Apriori. Program je spouštěn z příkazové řádky:

```
apriori log_file minsup minconf
```

Všechny tři argumenty jsou povinné:

`log_file` je cesta k souboru, kde je záznam uživatelských transakcí. Jeden řádek v souboru odpovídá jedné uživatelské relaci. Každá unikátní transakce je reprezentována celým číslem. Transakce patřící do jedné uživatelské relace jsou na jednom řádku seřazeny vzestupně od transakce s nejnižším číslem po transakci s nejvyšším číslem.

`minsup` udává hodnotu minimální podpory asociačních pravidel. Nalezená asociační pravidla budou mít podporu rovnu nebo nebo větší než tato hodnota. Jsou dvě možnosti jak minimální podporu zadat - buď absolutním číslem nebo procentuálně, pro odlišení procentuálního zadání se za číslo přidá znak %.

`minconf` udává hodnotu minimální spolehlivost asociačních pravidel. Platí zde totéž jako u minimální podpory. Opět lze zadat absolutním číslem nebo procentuálně.

Výsledkem programu `apriori` je výpis nalezených asociačních pravidel na standardní výstup. Může vypadat nějak takto: 5 dat tabulku

```
110 ->61 Sup=19 Conf=90.4762
107 ->61 Sup=3 Conf=75
61 92 ->96 Sup=4 Conf=57.1429
107 110 ->61 Sup=3 Conf=75
61 91 110 ->103 Sup=5 Conf=55.5556
61 91 103 110 ->101 Sup=3 Conf=60
```

Sekvenční vzory má na starost program `gsp`, který nad vstupními daty provádí algorutmus GSP. Jeho spouštění i výstup jsou podobné jako `apriori`. Má stejné spouštěcí parametry jako `apriori`:

```
gsp log_file minsup minconf
```

Rozdíl je ve formátu souboru `log_file`, kde stejně jako u `apriori` jeden řádek odpovídá jedné uživatelské relaci a každá transakce je reprezentována celým číslem. Transakce ovšem musí být seřazeny chronologicky podle toho, kdy je uživatel provedl. Pro parametry `minsup` a `minconf` platí totéž co u `apriori`. Oba programy `apriori` i `GSP` jsou napsány v jazyce C++.

#### Předzpracování dat

Vytvoření souborů se záznamy transakčních historií je úkolem dvou skriptů napsaných v perlu `trans.pl` a `props_trans.pl`. Skript je spouštěn:

```
perl trans.pl -type=typ_transakci -format=format_vystupu
```

Argument `type` určuje, jaké typy transakčních historií bude výstup obsahovat. Možné hodnoty jsou:

- `loads` - ve výstupu budou transakce načtení stránek
- `ajax` - ve výstupu budou transakce použití AJAXu
- `obj` - ve výstupu budou transakce manipulace s objekty
- `ud` - ve výstupu budou uživatelsky definované transakce
- `all` - ve výstupu budou všechny transakce

Argument `format` nastaví způsob formátování a vnitřní strukturu výstupu podle toho pro jaký účel je výstup používán. Možné hodnoty jsou:

- `human` - čitelný výstup
- `ar` - výstup pro mining asociačních pravidel (program `apriori`)
- `sp` - výstup pro mining sekvenčních vzorů (program `gsp`)
- `cf` - výstup pro kolaborativní filtrování (program `cf_suggest`)

Výstup skriptu `trans.pl` je vypisován na standardní výstup. Skript `props_trans.pl` vytváří transakční historii vlastností navštívených stránek. Spouští se:

```
perl props_trans.pl -format=format_vystupu -dir=adr_vystupu
```

Argument `format` je stejný jako u `trans.pl`, nastaví výstup podle toho, pro jaké zpracování je určen. Možné hodnoty jsou stejné.

Argument `dir` určuje adresář, kam budou zapsány soubory s transakčními historiemi jednotlivých vlastností. Pokud se například na stránkách webové aplikace objevují vlastnosti *author* a *category*, pak po spuštění skriptu budou v nastaveném adresáři soubory s transakčními historiemi `author.log` a `category.log`.

Parametr `type` u `props_trans.pl` není použit, protože transakční historie vlastností se týkají pouze transakcí načtených stránek. `props_trans.pl` na standardní výstup nic nevypisuje, neboť výstup je rovnou ukládán do souborů ve specifikovaném adresáři.

Oba skripty `trans.pl` i `props_trans.pl` čerpají data z databáze, jejich úkolem v rámci frameworku je předzpracování dat (viz část 2.1.2).

#### Crawler

Crawler je pomocný skript, který automaticky prochází stránky webové aplikace a zjišťuje hodnoty vlastností nastavené pomocí metatagu a také názvy stránek (tag `<title>`). Vlastnosti a názvy stránek ukládá do databáze. Crawler je implementován jako shell skript.

#### Spuštění programů *data mining modulu*

Zmíněné části *data mining modulu* nejsou spouštěny ručně, ale automatickým shell skriptem `mine.sh`. `mine.sh` spouští ve správném pořadí jednotlivé programy. Skript spouští jednotlivé programy s parametry, které jsou nastavené v konfiguračním souboru (příklad konfiguračního souboru ukázkové aplikace je v příloze). V něm má tvůrce webové aplikace možnost nastavit, které metody dobývání znalostí se použijí a na které typy transakcí se aplikují. Dají se v něm nastavit také hodnoty minimálních podpor a minimálních spolehlivostí pro jednotlivé způsoby dobývání znalostí.

### 5.2.3 SOAP server

Server je implementován pomocí kolekce modulů SOAP::Lite [22] napsané pro perl. Smyslem serveru je umožnit komunikaci mezi *interface modulem* frameworku

a *výpočetním modulem*. Rozhraní serveru nabízí několik funkcí, zde je jejich přehled:

`service_available` - funkce pro zjištění, zda je server online

`save_log` - funkce uloží záznam transakce uživatele do databáze a případně v odpovědi odešle klientovi výsledky analýzy uživatelovi transakční historie, má tři argumenty:

- `uid` - identifikátor uživatele
- `transaction` - informace o provedené transakci (typ,url adresa, doplňující textové informace, indikátor, zda se jedná o uživatelsky definovanou transakci)
- `compute_vector` - pokud je nastaven, funkce v odpovědi pošle výsledky analýzy; argument je vektor a určuje metody, které budou při analýze provedeny

`get_results` - funkce slouží jako žádost o provedení analýzy uživatelovi transakční historie, volá funkci *modulu pro analýzu transakcí* `compute_transaction`; má dva argumenty:

- `uid` - identifikátor uživatele
- `compute_vector` - stejný jako u `save_log`
- `transaction` - volitelný argument, ovlivňuje nalezené výsledky, pokud je takto zadaná transakce, tak výsledky metod hledání asociačních pravidel a sekvenčních ji používají jako povinnou transakci a v každém nalezeném pravidle (vzoru) musí být na levé straně pravidla (vzoru)

### 5.2.4 Modul pro analýzu transakcí

Úkolem tohoto modulu na vykonat online mining pro aktuálního uživatele a jeho transakční historii. Modul je napsán v perlu. Klíčové funkce jsou:

`compute_transaction(uid,compute_vector,transaction)` - funkce vybere z databáze veškeré transakce, které odpovídají identifikaci uživatele `uid` a vytvoří z nich transakční historie těch typů, kterých je potřeba; poté hledá odpovídající asociační pravidla, sekvenční vzory případně udělá kolaborativní filtrování, výsledky předává SOAP serveru; které transakčních historie budou po-

užity rozhoduje argument `compute_vector`; poslední parametr `transaction` určuje povinou transakci (viz funkce SOAP serveru `get_results`)

`get_transaction(uid,type)` - vyzvedne z databáze, transakční historii odpovídající identifikaci uživatele `uid`, typ transakční historie určuje argument `type`

# 6 Ukázková aplikace

Práce frameworku je demonstrována na webové aplikaci, která ničím nevybočuje ze standardů, podle nichž se dnešní webové aplikace vytvářejí. Ukazková aplikace je eshop nabízející zboží s výpočetní technikou. Eshop je napsaný v PHP, používá javascriptové dynamické prvky včetně AJAXu. V javscriptovém kódu je použita knihovna jQuery. Eshop s nasazeným frameworkem běží na adrese:

Struktura webové aplikace je poměrně jednoduchá. Aplikace slouží kromě eshopu jako prezentace společnosti provozující eshop. Web je tvořen několika málo informačními stránkami, kde jsou uvedeny kontakty společnosti nebo informace o kameném obchodu. Eshop je pouze doplňkovou obchodní aktivitou společnosti a nemá příliš vysokou návštěvnost. Framework používal data získaná z návštěv uživatelů během jednoho týdne. V tomto časovém úseku bylo zaznamenáno řádově pár stovek plnohodnotných návštěv, z nichž bylo možné čerpat.

Odkazy na informační stránky jsou v hlavním vertikálním menu. Navigace v samotném eshopu je v levém sloupci aplikace. Zde jsou odkazy na jednotlivé kategorie nabízeného zboží, alternativou je zobrazení odkazů nikoliv na kategorie zboží, ale na výrobce. Každá kategorie může mít další podkategorie (např. kategorie *Komponenty* má podkategorii *Zvukové karty*), ty jsou po zobrazení konkrétní kategorie přidány do navigace v levém menu.

Na stránce s detailem kategorie je pak zobrazen seznam zboží v kategorii i v podkategoriích. Na každé stránce je maximálně devět produktů, další produkty jsou dostupné pomocí stránkování umístěného v dolní části pod přehledem devíti produktů. Odkazy na stránkování nejsou skutečné odkazy nýbrž spouštěče skriptu, který nenačítá znova celou stránku, ale pouze další produkty k zobrazení, k tomu je použit AJAX. Na stránce lze měnit seřazení produktů, implicitní je seřazení podle názvu produktů, další možné seřazení je opačné podle názvu nebo podle ceny vzestupně a sestupně. Při změně seřazení je k novému zobrazení použit AJAX.

Z přehledu produktů se lze kliknutím na produkt přemístit na stránku s detailními informacemi o produktu. Kromě základních informacích o produktu je na této stránce rámeček s detailním popisem produktu, který lze dynamicky přepnout na graf ukazující vývoj ceny produktu. Dále je zde tlačítko umožňující přidání produktu do uživatelova nákupního košíku.

Obsah košíku je zobrazen v prostoru vpravo na každé stránce. Zde je také odkaz vedoucí do podrobnějšího přehledu obsahu košíku. Na stránce s obsahem košíku je ještě možnost potvrdit objednávku.

Poslední podstatnou součástí eshopu je jednoduché vyhledávání. Formulář pro vyhledávání je umístěn v pravé horní části každé stránky. Lze zde zadat frázi, jejíž výskyt je pomocí jednoduchého fulltextového vyhledávacího enginu hledán v názvech a v dalších informacích o produktech. Výsledek vyhledávání je zobrazen stejným způsobem jako přehled produktů v kategorii.

## 6.1 Transakce v aplikaci

V eshopu je možné sledovat veškeré typy transakcí, které framework umožňuje sledovat. Jak se ukáže, jen některé z nich má v tomto případě smysl zaznamenávat. Následuje přehled transakcí v aplikaci.

#### Načtení stránky

U načítání stránek je nutno podotknout, že v eshopu je občas namísto načtení celé stránky použito jen načtení části obsahu pomocí AJAXu. To se stává v přehledu produktů v kategorii při použití stránkování či při změně řazení produktů.

#### Použití AJAXu

AJAX je v eshopu použit jako doplňková technologie. Jak již bylo zmíněno využívá se u stránkování produktů a při změně řazení. Pro zobrazení obsahu stránky s produkty je nejprve AJAXem zavolán skript, který generuje XML s potřebnými informacemi o žádaných produktech a dále se pomocí AJAXu získá XSL šablona určující vzhled.

#### Manipulace s objekty

Eshop je kromě používání AJAXu poměrně statická aplikace a jediné další dynamické prvky jsou na stránkách detailů produktů. Dynamická je změna zobrazení popisu produktu na graf s vývojem ceny.

#### Uživatelsky definované transakce

V eshopu je použita jedna uživatelsky definovaná transakce, a to je přidání produktu do košíku.

#### Vlastnosti stránek

Každá stránka v eshopu má minimálně jednu a maximálně tři vlastnosti. Možné vlastnosti jsou:

**type** - slouží pro určení typu stránky, tato vlastnost je na všech stránkách, použité hodnoty jsou:

*product* - stránka s detailem produktu

*category* - stránka s přehledem produktů v kategorii

*info* - ostatní stránky

**category** - tuto vlastnost mají stránky typu *product* a *category*, je zde identifikační číslo kategorie, v jaké se produkt nachází či identifikační číslo samotné kategorie

**price** - vlastnost se objevuje pouze u stránek typu *product*, je to číslo od 0 do 10, které vyjadřuje cenu produktu v rámci kategorie, nejlevnější produkt v kategorii má hodnotu 0, naopak nejdražší má hodnotu 10

## 6.2 Vzorové nastavení frameworku

Konfigurační soubor *výpočetního modulu* frameworku pro eshop je ukázán v příloze. V konfiguračním souboru jsou nastaveny minimální hodnoty podpory a spolehlivosti metod hledání asociačních pravidel a sekvenčních vzorů a dále váhy přikládané výsledkům jednotlivých metod, nulová váha znamená, že metoda není použita. Minimální hodnoty i váhy se dají nastavit různě pro dobývání znalostí z různých transakčních historií. Například `as_rules_minsup_page` udává minimální podporu pro asociační pravidla v transakční historii, obsahující načtení stránek, `as_rules_minsup` je minimální podpora u celkové transakční historie. Vzhledem k relativně nízkému počtu zaznamenaných transakčních historií jsou hodnoty minimálních podpor nastaveny absolutními hodnotami a nikoliv v procentech.

Jak bylo zmíněno, zaznamenávají se všechny typy transakcí. Výsledky analýzi chování aktuálního uživatele se odesílají jen na požadavek, hodnota parametru `results_on_demand_only` je tedy `true`.

Ačkoliv se zaznamenávají všechny typy transakcí, výsledky analýzy z historií transakcí typu použití AJAXu a manipulace s objekty nejsou použity. Ani jedna z metod web usage mining v nich nenalezla žádné významné pravidelnosti. U AJAXu je to pochopitelné, neboť tvůrce webu použil technolgii AJAX jako pomocný prostředek pro zobrazování částí obsahu. Nalezené pravidelnosti v transakčních historiích použití AJAXu tedy vypovídají o záměrech tvůrce webové aplikace a ne o uživatelích. Co se týče transakcí typu manipulace s objekty, tak tam je situace obdobná. Manipulace se využívá jako doplňková funkce a bývá zaznamenána zřídka.

## 6.3 Cílové adaptace aplikace

Byly navrženy tři způsoby jak využívat výsledky analýzi frameworku. Jejich cílem je nabídnout zákazníkovi rychlejší a přímější přístup k produktům, které by ho mohly zajímat. Adaptace byly vybrány s ohledem na dvě věci. Za prvé aby adaptace byly smysluplné, tedy nabízeli návštěvníkům užitečné informace. A za druhé, aby na nich mohly být demonstrováno použití různých typů transakčních historií a metod dobývání znalostí.

### Podobné produkty

První z interpretací výsledků analýzi frameworku je adaptace stránky s detailem produktu. Cílem je zobrazit produkty, které jsou podobné tomu, který je na stránce představován. Podobnost produktů je v tomto případě určováno podle toho, o které z nich projevili uživatelé zájem během jedné návštěvy eshopu, tedy navštívili stránku s detailními informacemi o produktu.

### Podobné kategorie

Tato adaptace je obdobná jako předchozí, zaměřuje se na propagování kategorií produktů, které bývají navštěvovány během jedné relace.

### Další kroky

Poslední adaptace je zaměřena na celkovou transakční historii. Pokud se uživatel dostane do detailu produktu, je mu nabídnut přehled transakcí, které posléze vykonali předchozí návštěvníci po zobrazení tohoto produktu.

## 6.4 Praktické ukázky adaptací

V této kapitole se budu věnovat praktickému předvedení analýzi *výpočetního modulu* a ukázky její interpretace pomocí zmíněných adaptací.

### Podobné produkty

K hledání podobných produktů je u metod web usage mining použita transakční historie načtených stránek. Nejprve nastavíme *výpočetní modul*, aby použil všechny tři metody web usage mining, tak jak je to v přiloženém konfiguračním souboru. Prahy spolehlivosti a podpory asociačních pravidel a sekvenčních vzorů jsou shodně nastaveny na 5% u spolehlivosti a na 3 u podpory:

```
as_rules_minconf_page=5%
as_rules_minsup_page=3
sqp_rules_minconf_page=5%
sqp_rules_minsup_page=3
```

Při testování byla použita stránka s procesorem *INTEL Core i7 920*. Výsledkem pak na stránce s detailem procesoru je tento seznam produktů:

```
INTEL Core i7 940 2.93GHz, 8MB, QPI 4,8GT, socket 1366, BOX
INTEL Core 2 Duo E8600 3.33GHz, 6MB, 1333MHz, socket 775, BOX
INTEL Core i7-975 Extreme 3.33GHz, 8MB, QPI 6,4GT, soc. 1366, BOX
INTEL Core 2 Quad Q9550 2.83GHz, 12MB, 1333MHz, socket 775, BOX
ASUS P6T, s.1366, iX58, 6xDDR3, 3xPCIE, SATA3Gb/sx6
INTEL Core 2 Quad Q8200 2.33GHz, 4MB, 1333MHz, socket 775, BOX
INTEL Core i7-965 Extreme 3.20GHz, 8MB, QPI 6,4GT, soc. 1366, BOX
INTEL Core 2 Quad Q8400 2.66GHz, 4MB, 1333MHz, socket 775, BOX
INTEL Core i7 950 3,06GHz, 8MB, QPI 4,8GT, socket 1366, BOX
INTEL Core 2 Duo E8500 3.16GHz, 6MB, 1333MHz, socket 775, BOX
GIGABYTE s775 EP45-UD3P, Intel P45, Ultra Durable 3
Intel Skyberg DP45SG, s.775, P45, DDR3, GLAN, ATX
Intel Bonetrail 2 DX48BT2, s.775, X48, 4xDDR3, PCIE, GLAN, ATX
ASUS P5Q, s.775, PCI-E, P45, SATA 3Gb/s*6, ATX
```

Pokud se použijí jednotlivé metody samostatně, pak je výstup u každé metody různý.
Podobné produkty pouze pomocí asociačních pravidel:

```
INTEL Core i7 940 2.93GHz, 8MB, QPI 4,8GT, socket 1366, BOX
INTEL Core 2 Duo E8600 3.33GHz, 6MB, 1333MHz, socket 775, BOX
INTEL Core i7-975 Extreme 3.33GHz, 8MB, QPI 6,4GT, soc. 1366, BOX
INTEL Core 2 Quad Q9550 2.83GHz, 12MB, 1333MHz, socket 775, BOX
ASUS P6T, s.1366, iX58, 6xDDR3, 3xPCIE, SATA3Gb/sx6
INTEL Core 2 Quad Q8200 2.33GHz, 4MB, 1333MHz, socket 775, BOX
INTEL Core i7-965 Extreme 3.20GHz, 8MB, QPI 6,4GT, soc. 1366, BOX
INTEL Core 2 Quad Q8400 2.66GHz, 4MB, 1333MHz, socket 775, BOX
INTEL Core i7 950 3,06GHz, 8MB, QPI 4,8GT, socket 1366, BOX
INTEL Core 2 Duo E8500 3.16GHz, 6MB, 1333MHz, socket 775, BOX
GIGABYTE s775 EP45-UD3P, Intel P45, Ultra Durable 3
Intel Skyberg DP45SG, s.775, P45, DDR3, GLAN, ATX
Intel Bonetrail 2 DX48BT2, s.775, X48, 4xDDR3, PCIE, GLAN, ATX
ASUS P5Q, s.775, PCI-E, P45, SATA 3Gb/s*6, ATX
```

Podobné produkty pouze pomocí sekvenčních vzorů:

```
INTEL Core i7 940 2.93GHz, 8MB, QPI 4,8GT, socket 1366, BOX
INTEL Core i7-965 Extreme 3.20GHz, 8MB, QPI 6,4GT, soc. 1366, BOX
INTEL Core i7 950 3,06GHz, 8MB, QPI 4,8GT, socket 1366, BOX
INTEL Core i7-975 Extreme 3.33GHz, 8MB, QPI 6,4GT, soc. 1366, BOX
Intel Bonetrail 2 DX48BT2, s.775, X48, 4xDDR3, PCIE, GLAN, ATX
GIGABYTE s775 EP45-UD3P, Intel P45, Ultra Durable 3
Intel Skyberg DP45SG, s.775, P45, DDR3, GLAN, ATX
```

Podobné produkty pouze pomocí kolaborativního filtrování:

```
INTEL Core 2 Quad Q8400 2.66GHz, 4MB, 1333MHz, socket 775, BOX
INTEL Core i7 940 2.93GHz, 8MB, QPI 4,8GT, socket 1366, BOX
```

#### Podobné kategorie

Zobrazování podobný kategorií je umožněno na stránce s přehledem produktů v určité kategorii. Podobné kategorie jsou umístěny napravo od seznamu produktů. Metody web usage mining v tomto případě používají transakční historii vlastnosti `category`. Hodnoty minimální podpory a minimální spolehlivosti jsou opět stejné u asociačních pravidel i sekvenčních vzorů:

```
p_as_rules_minconf=10%
p_as_rules_minsup=3
p_sqp_rules_minconf=10%
p_sqp_rules_minsup=3
```

Použita byla stránka kategorie *Procesory*, výsledné podobné kategorie jsou tyto:

```
Komponenty
Základní desky
Socket 775
Socket AM2
Pameti
DDR 3
Grafické karty
Pevné disky a rámečky
Pevné disky
Socket 1366
Socket AM3
DDR 2
```

Podobné kategorie pouze pomocí asociačních pravidel:

```
Komponenty
Základní desky
Socket 775
Socket AM2
Pameti
DDR 3
Grafické karty
Pevné disky a rámečky
Pevné disky
Socket 1366
Socket AM3
DDR 2
```

Podobné kategorie pouze pomocí sekvenčních vzorů:

```
Komponenty
Základní desky
Socket AM2
Pameti
DDR 3
Pevné disky a rámečky
Pevné disky
DDR 2
```

Podobné kategorie pouze pomocí kolaborativního filtrování:

```
Komponenty
Základní desky
Socket AM2
Pameti
DDR 3
Pevné disky a rámečky
Pevné disky
Socket 1366
Socket AM3
DDR 2
```

### Další kroky

Ke zjišťování dalších možných kroků uživatele jsou použity pouze asociační pravidla, která pracují s celkovou transakční historií. Minimální hodnota spolehlivosti je nastavena na 0:

```
as_rules_minconf=0
as_rules_minsup=3
```

Doporučené další kroky na stránce s detailem procesoru *INTEL Core i7 920* jsou tyto:

```
27% uživatelů, kteří navštívili tuto stránku si koupilo INTEL Core i7 920
20% uživatelů, kteří navštívili tuto stránku si koupilo ASUS P6T, s.1366, iX58,
6xDDR3, 3xPCIE, SATA3Gb/sx6
```

# 7 Závěr

Za hlavní přínos práce považuji to, že byla ukázána možnost použití složitějších metod personalizace pro libovolné webové aplikace. Netriviální metody, které jsou v dnešní době implementovány převážně ve vysokorozpočtových komerčních aplikacích, můžou být pomocí frameworku použity téměř v jakékoliv webové aplikaci. Framework poskytuje své funkce v podobě webové služby a díky tomu nejsou na straně samotné webové aplikace potřeba žádné speciální technologie.

V ukázkové aplikaci bylo demonstrováno, že výsledky práce frameworku mohou být pro webové aplikace v komerční oblasti vhodným doplňkem pro prezentaci informací a produktů. V eshopu je vidět, že adaptace provedené na základě doporučení frameworku dávají smysl a adaptacemi propagované produkty mají souvislost s tím, které stránky uživatel prohlíží a můžou pro něj být nezanedbatelným přínosem. Důležitá je možnost výběru metod web usage mining, které mohou být použity. V ukázkové aplikaci aplikaci se jako nejlepší metoda jeví získávání asociačních pravidel, nejslabší naopak je použití sekvenčních vzorů. To je do jisté míry dáno povahou aplikace, kde příliš nehraje roli v jakém pořadí udělá uživatel své kroky u jiných webových aplikací může být hodnocení kvality metod opačné.

V porovnání s obdobnými projekty zmíněnými v úvodu, je přínos práce v tom, že umožňuje aplikovat metody personalizace v libovolné webové aplikaci a nezávisí na tom jaký je účel existence aplikace. Je ponecháno na tvůrci webové aplikace, aby se rozhodl, zda a které metody personalizace poskytované frameworkem jsou pro něj vhodné. Existující projekty v oblasti personalizace jsou obvykle zaměřeny na konkrétní webovou aplikaci případně na určitý typ webových aplikací, tím je jejich možnost použití omezená.

## 7.1 Možná vylepšení

O současné podobě frameworku lze říci, že základním způsobem plní svoji funkci. Nabízí se řada způsobů jak zpřesnit či vylepšit práci frameworku.

Jednou z možností vylepšení, která by mohla mít pozitivní přínos je rozšířit sledování chování uživatelů. Uvažovat by se dalo o použití vah pro jednotlivé transakce, které uživatel provede. To lze aplikovat například u transakcí typu načtení

stránky. Doba, kterou uživatel na stránce stráví, indikuje míru zájmu o informace na stránce. Transakce načtení stránky, kde uživatel stráví nejvíce času, by tedy měla nejvyšší váhu. Použití vah by mohlo nabídnout zajímavé výsledky hlavně při použití kolaborativního filtrování.

Další zpřesnění sledování uživatelů by se mělo týkat opakovaných návštěv uživatelů. Nyní framework pracuje pouze s aktuální relací uživatele a jeho předchozí návštěvy v úvahu nebere.

Framework rovněž nepracuje s informacemi získanými z browseru uživatele či informacemi získanými explicitně z dotazníků nebo formulářů ve webové aplikaci. Použití těchto dat by jistě mohlo mít pozitivní efekt.

Další věc, která ve frameworku není implementována je pozorování dopadu adaptací na uživatele. U některých adaptací je po jejich provedení možné určit, zda jejich přínos pro uživatele byl pozitivní nebo negativní. Pokud například adaptace frameworku vygeneruje odkaz na stránku, uživatel jej použije a následně krátce po navštívení této stránky klikne v browseru na tlačítko "Zpět", pak adaptaci lze označit jako chybnou a framework by tutu zkušenost měl v budoucnu brát v úvahu.

# Literatura

[1] Apostolou D., Feldkamp D., Halaris C., Hinkelmann K., Magoutas B., Papadomichelaki X., Prackwieser C., Probst F., Schmidt K.U., Stoiljkovic B., Stoiljkovic V., Stojanovic L., Stojanovic N., Thomas S. M., Thönssen B., Utz W., Woitsch R. *IST PROJECT 27090 Fostering self-adaptive e-government service improvement using semantic technologies*, 2006

[2] Robert Cooley, Bamshad Mobasher, Jaideep Srivastava *Web Mining: Information and Pattern Discovery on the World Wide Web. In Proceedings of the 9th IEEE International Conference on Tools with Artifi cial Intelligence*, ICTAI'97, 1997

[3] Raymond Kosala, Hendrik Blockeel *Web Mining Research: A Survey. SIGKDD Explorations*, 2000

[4] Bettina Berendt, Andreas Hotho and Gerd Stumme *Towards Semantic Web Mining*, 2002

[5] Gregory S. Barnes Nelson *Avoiding eOverload: Personalizing Web Content through Security, eIntelligence and Data Mining*, 2001

[6] Robert Cooley, Bamshad Mobasher, Jaideep Srivastava *Data Preparation for Mining World Wide Web Browsing Patterns, Knowledge and Information Systems*, 1999

[7] *http://www.w3.org/Daemon/User/Config/Logging.html*

[8] *http://www.google.com/analytics/*

[9] Bing Liu *Web Data Mining*, 2007

[10] Alfred Kobsa, Jürgen Koenemann, Wolfgang Pohl *Personalized techiques for improving online customer relationships*, 2001

[11] J. Ross Quinlan *C4.5: Programs for Machine Learning. Morgan Kaufmann*, San Mateo, CA, 1993

[12] Bamshad Mobasher, Robert Cooley, Jaideep Srivastava *Automatic Personalization Based on Web Usage Mining*, 2003

[13] Josef Fink, Alfred Kobsa *Adaptable and Adaptive Information Provision for All Users, Including Disabled and Elderly People*, 1998

[14] Paul De Bra, Licia Calvi () *AHA: a Generic Adaptive Hypermedia System*

[15] *http://www.fit-project.org/*

[16] Rakesh Agrawal, Tomasz Imielinski, Arun Swami *Mining Association Rules between Sets of Items in Large Databases*, 1993

[17] *http://www.w3.org/TR/2006/WD-XMLHttpRequest-20060405/*

[18] *http://dev.w3.org/2006/waf/access-control/*

[19] *https://developer.mozilla.org/En/Same_origin_policy_for_JavaScript*

[20] *http://json.org/*

[21] *http://jquery.com/*

[22] *http://www.w3.org/TR/soap/*

[23] Greg Linden, Brent Smith, Jeremy York *Amazon.com Recommendations: Item-to-Item Collaborative Filtering*, 2001

[24] George Karypis *Evaluation of Item-Based Top-N Recommendation Algorithms*, 2001

[25] *http://www.soaplite.com/*

[26] Rakesh Agrawal, Ramakrishnan Srikant *Fast Algorithms for Mining Association Rules*, 1994

# Přílohy

# Dodatek A - Konfigurační soubor webové aplikace

```
# nastaveni minimalnich hodnot pro asociacni pravidla
# celkove transakcni historie
as_rules_minconf=0
as_rules_minsup=3

# nastaveni minimalnich hodnot pro asociacni pravidla
# transakcni historie obsahujici nactene stranky
as_rules_minconf_page=5%
as_rules_minsup_page=3

# nastaveni minimalnich hodnot pro asociacni pravidla
# transakcni historie pouziti ajaxu
as_rules_minconf_ajax=10%
as_rules_minsup_ajax=55%

# nastaveni minimalnich hodnot pro asociacni pravidla
# transakcni historie manipulace s objekty
as_rules_minconf_obj=10%
as_rules_minsup_obj=4

# nastaveni minimalnich hodnot pro asociacni pravidla
# transakcni historie s uzivatelsky
# definovanymi udalostmi
as_rules_minconf_ud=10%
as_rules_minsup_ud=3

# nastaveni minimalnich hodnot pro asociacni pravidla
# transakcni historie vlastnosti
p_as_rules_minconf=10%
```

```
p_as_rules_minsup=3

# nastaveni minimalnich hodnot pro sekvencni vzory
# celkove transakcni historie
sqp_rules_minconf=10%
sqp_rules_minsup=7

# nastaveni minimalnich hodnot pro sekvencni vzory
# transakcni historie obsahujici nactene stranky
sqp_rules_minconf_page=5%
sqp_rules_minsup_page=3

# nastaveni minimalnich hodnot pro sekvencni vzory
# transakcni historie pouziti ajaxu
sqp_rules_minconf_ajax=10%
sqp_rules_minsup_ajax=15%

# nastaveni minimalnich hodnot pro sekvencni vzory
# transakcni historie manipulace s objekty
sqp_rules_minconf_obj=10%
sqp_rules_minsup_obj=4

# nastaveni minimalnich hodnot pro sekvencni vzory
# transakcni historie s uzivatelsky
# definovanymi udalostmi
sqp_rules_minconf_ud=10%
sqp_rules_minsup_ud=4

# nastaveni minimalnich hodnot pro sekvencni vzory
# transakcni historie vlastnosti
p_sqp_rules_minconf=10%
p_sqp_rules_minsup=3
```

```
# vaha asociacnich pravidel
association_rules_weight=1

# vaha kolaborativniho filtrovani
collaborative_filtering_weight=1

# vaha sekvencnich vzoru
sequential_patterns_weight=1

# vaha asociacnich pravidel pro uzivatelsky
# definovane transakce
ud_association_rules_weight=1

# pro celkove transakce se pouzivaji
# jen asociacni pravidla
all_collaborative_filtering_weight=0
all_sequential_patterns_weight=0
```

# Dodatek B - Instalace frameworku

### Instalace interface modulu

*Interface modul* se přidává přímo ke kódům webové aplikace. Je potřeba aby na http serveru, kde webová aplikace běží, bylo k dispozici PHP.
Postup instalace:

1. rozbalit archiv `/interface_modul/sp_int_modul.bz2` z CD do kořenového adresáře webové aplikace, vznikne zde nový adresář `sp_pack`
2. nastavit adresu *výpočetního modulu* - v souboru `sp_pack/config.php` nastavit hodnotu proměnné `soa_server_address`
3. přidat inicializační funkci frameworku `sp.init()` (viz kapitola 5.1) do hlavičky na všech stránkách webové aplikace

### Instalace výpočetního modulu

*Výpočetní modul* je vytvořen pro unixovské operační systémy. Pro správnou funkci *výpočetního modulu* je potřeba mít nainstalován http server, MySQL databázi, interpret jazyka Perl a modul SOAP::Lite.
Postup instalace:

1. rozbalit archiv `/vypocetni_modul/sp_comp_modul.bz2` z CD do vybraného adresáře, archiv obsahuje předkompilované binární soubory některých programů frameworku pro systémy Linux x86, pro jiné systémy je potřeba tyto programy překompilovat:

   ```
   cd inst_addr/data_process/apriori/
   make
   cd inst_addr/data_process/cf/
   make
   cd inst_addr/data_process/gsp/
   make
   ```

   kde `inst_addr` je adresář, kam byl rozbalen celý archiv.
2. do zvolené MySQL databáze importovat inicializační skript `/vypocetni_modul/sp_sql.bz2`.

3. nastavit http server tak, aby soubor `inst_addr/soap-nn/process.cgi/` byl přístupný pomocí protokolu http.
4. provést konfiguraci v souboru `inst_addr/cfg/sp.cfg`:
   - nastavit správně cesty k jednotlivým komponentám *výpočetního modulu*
   - nastavit přístupové informace k MySQL databázi (host - `mysql_host`, login jméno - `mysql_user`, heslo - `mysql_pass`, název databáze - `mysql_db`)

# Dodatek C - Návod k použití frameworku

Chování frameworku lze ovlivnit nastavením *interface modulu* a zároveň i úpravou konfigurace *výpočetního modulu*. Nastavení v obou modulech jsou různá.

### Nastavení interface modulu

Aby framework správně fungoval musí být na každé stránce v hlavičce kromě odkazu na javascriptový kód *interface modulu* inicializační funkce `sp.init()`. Pomocí jejích parametrů pak lze nastavit jaké akce uživatele bude framework zaznamenávat a jak se budou zpracovávat výsledky analýzy frameworku. Podrobný přehled parametrů funkce `sp.init()` je v kapitole 5.1. Inicializace může vypadat takto:

```
<script type="text/javascript"
        src="http://app_url/sp_pack/sp.js"></script>
<script type="text/javascript">
  sp.init({
    log_settings : "all",
    compute_loads: true,
    compute_ajax : false,
    compute_obj : false,
    compute_ud : false,
    compute_properties : true,
    compute_all : true,
    adapt_loads_handler : fx_loads,
    adapt_properties_handler : fx_props,
    adapt_all_handler : fx_all,
    results_on_demand_only : true
  });
</script>
```

Tato inicializace nastavuje, že logovány budou veškeré možné typy akcí uživatele (`log_settings : "all"`), kterými jsou načtení stránek, použití AJAXu i manipulace s objekty. Dále hodnoty `compute_loads`, `compute_properties`

a `compute_all` jsou nastaveny na `true`, což říká *výpočetnímu modulu*, aby prováděl 3 různé analýzy: na transakční historii s načteními stránek, na transakční historii vlastností a na celkovou transakční historii. Výsledky jednotlivých analýz budou předány funkcím určenými parametry (`adapt_loads_handler`, `adapt_properties_handler`, `adapt_all_handler`). Formát výsledků předaný těmto funkcím je javascriptové pole, ve kterém každá položka je objekt reprezentující jednu doporučenou transakci. Jedna položka má tyto části:

`type` - typ transakce

`url` - nemusí být použit vždy, u transakce typu načtení stránky obsahuje adresu stránky

`title` - nemusí být použit vždy, u transakce typu načtení stránky obsahuje název stránky

`obj` - nemusí být použit vždy, u uživatelsky definované transakce obsahuje parametr transakce; u transakce vlastnosti obsahuje hodnotu vlastnosti

`weight` - reálné číslo uvádějící kvalitu předpovědi transakce

Javascriptová funkce zpracovávající analýzu transakce načtených stránek, pak může vypadat takto:

```
function fx_loads(eloads) {
    $("#similar").prepend("Doporučené");
    for (var i = 0; i < eloads.length; i++)
        $("#similar").append('<a href="'+eloads[i].url+
          '">'+eloads[i].title+'</a>');
}
```

O provedení analýzi lze zažádat javascriptovou funkcí `get_user_results()` nebo `get_user_results_ctrans()`, jejich popis je v kapitole 5.1.1.

K zaznamenávání uživatelsky definovaných událostí slouží funkce `log_ud_event()`, popis je opět v kapitole 5.1.1.

Pokud se sledují vlastnosti stránek, tedy `compute_properties` je v `sp.init()` nastaveno na `true`, tak by se na určitých stránkách měly v hlavičce objevovat metatagy s vlastnostmi. Viz kapitola 5.1.3. Hodnoty vlastností by měly být diskrétní. Vlastnosti se nemusí vyskytovat na všech stránkách, stačí, aby byly jen na některých.

Je jasné, že framework nebude poskytovat doporučení hned od počátečního nasazení. Framework nejprve musí získat vzory chování dostatečného počtu uživatelů webové aplikace. V první fázi používání frameworku tedy nejsou k dispozici žádné výsledky, ale probíhá pouze sběr dat.

### Nastavení výpočetního modulu

Výpočetní modul je nastavován výhradně konfiguračním souborem `cfg/sp.cfg`. Nastavují se zde:

- parametry metod dataminingu
- které metody se budou provádět
- na které typy transakčních historií se metody provádějí

Ukázka konfiguračního souboru, včetně popisu jednotlivých nastavení je v Dodatku A.

Příprava dat pro potřeby analýzi *výpočetního modulu* se dělá ručně skriptem `data_process/mine.sh`:

```
bash data_process/mine.sh -c -d
```

Tímto způsobem se vygenerují asociační pravidla a sekvenční vzory a dále se provede předzpracování dat pro potřeby kolaborativního filtrování. Skript `data_process/mine.sh` má smysl volat pouze v případě, že bylo sesbíráno dostatečné množství dat od uživatelů. Přepínač `-d` umožňuje provedení nutných zápisů do databáze *výpočetního modulu*. Přepínač `-c` spustí crawler. Nedoporučuje se spouštět skript bez těchto parametrů.

# Dodatek D - Struktura přiloženého CD

`/interface_modul` - zdrojové kódy *interface modulu*

`/sample_app` - zdrojové kódy ukázkové aplikace

`/text` - text diplomové práce a zdrojové kódy práce v TeXu

`/utils` - programy potřebné k fungování frameworku (SOAP::Lite, Apache HTTP Server, MySQL, PHP, Perl)

`/vypocetni_modul` - zdrojové kódy *výpočetního modulu*