

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

DATAMINING Z JABBERU

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAROSLAV SENDLER

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

DATAMINING Z JABBERU

DATAMINING FROM JABBERU

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JAROSLAV SENDLER

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JOZEF MLÍCH

BRNO 2011

Abstrakt

Předmětem této bakalářské práce bylo seznámení se s problematikou komunikace přes Jabber síť, která zde byla rozebrána. Konkrétním cílem bylo vytvoření jednoduchého Jabberového klienta, který by byl schopen získávat statistická data. Nashromážděná data sloužila pro pozdější analýzu a grafickou reprezentaci informací z nich získaných.

Abstract

The objective of this thesis was acquaint oneself with problems of communication via Jabber network, which was also analyzed. The specific objective was to create a simple Jabber's client which would be able to obtain statistical data. The collected data was used for analysis and graphic representation of information.

Klíčová slova

Jabber, XMPP, robot, datamining, dolování dat.

Keywords

Jabber, XMPP, robot, datamining.

Citace

Jaroslav Sendler: Datamining z jabberu, bakalářská práce, Brno, FIT VUT v Brně, 2011

Datamining z jabberu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Josefa Mlícha. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Jaroslav Sendler

14. dubna 2011

Poděkování

Tímto bych chtěl poděkovat mému vedoucímu bakalářské práce Ing. Josefovi Mlíchovi za ochotu a kladný přístup při konzultacích. Dále za poskytnutí hardware na němž běžel program a sbíral data.

© Jaroslav Sendler, 2011.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	2
2	XMPP	3
2.1	Architektura	3
2.2	XML	5
2.3	Stanza	6
2.4	Rozšíření	8
2.5	Knihovny	9
3	Data mining	11
3.1	Metody dolování dat	13
3.2	Shlukování	15
3.3	Programy	18
4	Developer	19
4.1	Databáze	19
4.2	Bot	19
4.3	Knihovny, jazyk	19
4.4	Jiné produkty	19
5	Vyhodnocení výsledků	21
6	Závěr	22
A	Obsah CD	25
B	Manual	26
C	Konfigurační soubor	27
D	Slovník výrazů	28
E	Stanza - základní schéma	29
E.1	iq	29
E.2	Message	30
E.3	Presence	30
E.4	Přehled průběhu rozšíření	31
F	Přehled klientů a jejich rozšíření	33

Kapitola 1

Úvod

Dnes mezi velmi se rozšiřující technologie na poli síťo

Kapitola 2

XMPP

Pro usnadnění a jednodušší pochopení budou v následující kapitole rozebrány základní stavební kameny protokolu Extensible Messaging and Presence Protocol (XMPP). Konkrétně jsou zde popsány stávající vlastnosti implementace [odkud se čerpalo], architektura protokolu XMPP obecně [17, 18] a další detaily protokolu [1, 19, 13]. Ty se vztahují k požadavkům na data mining popisovaný v této práci [16, 12]. Další informace byly čerpány z [20, 11, 4, 10, 9].

Samotný protokol XMPP je datován do roku 2004 (březen), kdy na něj byl přejmenován Jabber. Původní projekt Jabber byl vytvořen roku 1998 autorem Jeremie Miller, který ho založil na popud nesvobodných uzavřených IM služeb. Měl obsahovat tři základní vlastnosti -jednoduchost a srozumitelnost pro implementaci, jednoduchost rozšiřitelnosti a otevřenost. Základní vlastnosti a výhody klientů a serverů budou popsány níže. Roku 1999, 4.ledna byl vytvořen první server se jménem Jabber. Komunita vývojářů se chopila iniciativy a vytvořila klienty pro různé platformy (Linux, Macintosh, Windows), kteří dokázali se serverem komunikovat. Roku 2004 byl přidán mezi RFC (request of comments - žádost o komentáře) dokumenty. Základní normy jsou RFC 3920 (obecná specifikace protokolu) a RFC 3921 (samotný instant messaging a zobrazení stavu). Další zdokumentovaná rozšíření jsou vydávána v podobě tzv. XEP (XMPP Extension Protocol) dokumentů, starším jménem JEP (Jabber Enhancement Proposal). Dnešní počet těchto norem se blíží k číslu 300. Každý XEP obsahuje status, stav vývoje (schválení), ve kterém se zrovna nachází. Jako bezpečnostní prvky jsou zde podporovány SASL, TLS a GPG. XMPP protokol je postaven na obecném značkovacím jazyce XML, proto vlastnosti popsané v kapitole 2.2 na straně 5 platí i pro tento protokol.

2.1 Architektura

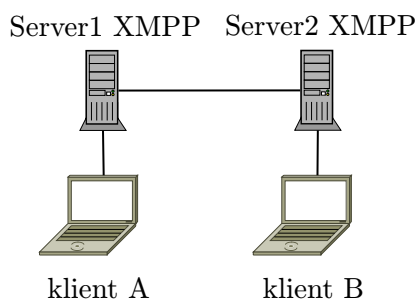
Dobře navržená architektura tvoří základ pro správně fungující internetovou technologii. XMPP protokol využívá decentralizované klient-server složení. Tato struktura se nejvíce podobá struktuře posílání e-mailů. V tomto případě je decentralizace sítě chápána jako inteligentní nezávislost mezi vývojáři klientů a serverů. Každý z nich má možnost se zaměřit na důležité části svého vývoje. Server na spolehlivost a rozšiřitelnost, klient na uživatele. Každý server pracuje samostatně, chod ani výpadek jiné datové stanice žádným způsobem neovlivní jeho běh, pouze bude nedostupný seznam kontaktů a služby, kterými server disponuje.

V tabulce č. 2.1 jsou shrnuty rozdíly v architektuře Jabber, WWW a e-mail¹. S každou zde jmenovanou službou má Jabber něco společného. Co se týče charakteristiky se vydal střední cestou. Na rozdíl od e-mailu, nepoužívá vícenásobný hops a v porovnání s WWW využívá mezi-doménové připojení.

Charakteristika	WWW	Email	Jabber
mezi-doménové připojení	Ne	Ano	Ano
vícenásobný hops	N/A	Ano	Ne

Tabulka 2.1: Srovnání služeb WWW, Email a Jabber

Tyto vlastnosti jsou zárukou pro bezpečný přenos zpráv, znemožňují "krádeže" JID², který je popsán v podkapitole Jabber ID 2.3, a spamování. Obrázek 2.1 zobrazuje přenos zprávy mezi klientem A, jehož účet vlastní *server1*, a klientem B s účtem na *serveru2*.



Obrázek 2.1: Přenos zprávy

Klient

Klient je především plně ovládatelný grafický program podporující jednoduché odesílání zpráv. V této práci je však zastupován robotem s konzolovým rozhraním. XMPP svou architekturou vnucuje, aby byl co nejjednodušší. Vlastnosti, které by měl mít jsou shrnuty do tří bodů:

1. komunikace se serverem pomocí TCP socketu
2. rozparsování a následná interpretace příchozí XML zprávy „stanza“ (kapitola 2.3)
3. porozumění sadě zpráv z Jabber jádra

Server

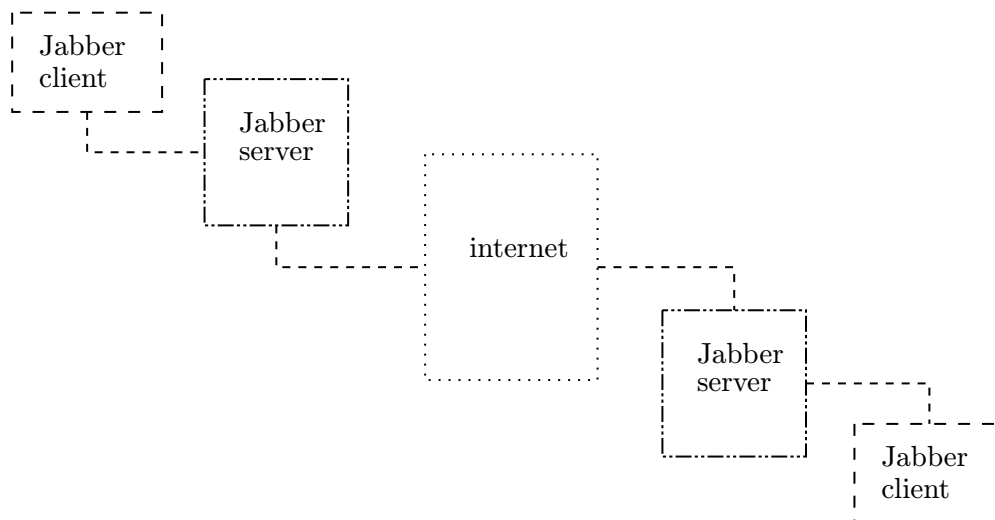
Hlavní vlastnost již není, jako u klienta, jednoduchost, ale stabilita a bezpečnost. Standardně běží na TCP portu 5222. Komunikace mezi servery je realizována přes port 5269. Každý server uchovává seznam zaregistrovaných uživatelů, kteří se do sítě mohou připojovat pouze přes něj. Tento seznam nemá žádný jiný server. To zajišťuje nemožnost „krádeže“

¹internetový systém elektronické pošty

²uživatelské jméno

účtu. Protože XMPP komunikace probíhá přes síť, musí mít každá entita adresu, v tomto případě nazvána JabberID. XMPP spoléhá na DNS což znamená že používá jména na rozdíl od IP protokolu.

Obrázek 2.2 znázorňující distribuovanou architekturu Jabberu byl převzat z [13].



Obrázek 2.2: Distribuovaná architektura Jabberu

V tabulce 2.2 jsou shrnuty informace o serverech Jabberu. První sloupec tvoří jméno, následuje programovací jazyk v němž je napsán. Většina je vydávána pod licenci GPL³, kromě ejabbred a Prosody. Ejabbred používá GPLv2, což je GPL licence druhé verze a Prosody licenci MIT/X11. U všech software byla zkoumána nejaktuálnější verze. Její číslo naleznete ve třetím sloupci. Všechny servery lze provozovat na operačním systému Linux a Windows. Na platformě Mac OS mohou být použity všechny zde jmenované vyjma jabberd2. Pět z šesti zde představených software pro server Jabber jsou ve stále vyvíjeny, tedy kromě jabberd14. Tabulka taktéž shrnuje důležité vlastnosti serverů v oblasti podpory rozšířených statusů o standardy XEP-0060 a XEP-0163 viz podkapitola 2.4.

Server	Jazyk	Verze	XEP-0060	XEP-0163
ejabberd	Erlang/	2.1.6	ANO	ANO
	Top			
Openfire	java	3.6.4	ANO	ANO
jabbred2	c	2.2.11	NE	NE
jabbred14	c,	1.6.1.1	ANO ⁴	NE
	c++			
Prosody	lua	0.7.0	NE ⁵	ANO
Tigase	java	5.0.0	ANO	ANO

Tabulka 2.2: Přehled Jabber serverů

³General Public License-všeobecná veřejná licence GNU

2.2 XML

Jazyk XML (eXtensible Markup Language), metajazyk pro deklaraci strukturovaných dat, je jádrem protokolu XMPP. Samotný jazyk vznikl rozšířením metajazyka SGML, jež slouží na deklaraci různých typů dokumentů. Základní vlastností je jednoduchá definice vlastních značek (tagů). Dokument XML se skládá z elementů, jež můžeme navzájem zanořovat. Vyznačujeme je pomocí značek - počáteční a ukončovací. Pomocí tohoto jazyka je tvořena stanza popsaná v kapitole 2.3.

Základní struktura dokumentu psaného jazykem XML je ukázána na obrázku 2.3. Každý dokument začíná XML deklarací a informací o kódování, ve kterém je dokument psán (1.řádek obrázku). Následuje kořenový element, jež je uzavřen na samotném konci dokumentu. 4. řádek ukazuje možnost použití prázdného elementu, který obsahuje jeden atribut s názvem zkratky fakulty.

```
1. <?xml version="1.0" encoding="utf-8"?>           // XML deklarace, kódování
2. <fakulta>                                           // kořenový element
3.   <název>Fakulta informačních technologií</název> // obsah elementu název
4.   <zkratka fakulty="FIT"/>                         // prázdný element
5.   <typy studia>                                     // počáteční tag
6.     <bakalářské titul="Bc."></bakalářské>          // název a hodnota atributu
7.     <magisterské></magisterské>
8.     <doktorské></doktorské>
9.   </typy studia>                                   // ukončovací tag
10.</fakulta>
```

Obrázek 2.3: Příklad základního XML dokumentu.

2.3 Stanza

Základní jednotkou pro komunikaci založenou na XML je stanza. Skládá se ze tří elementů *message*, *presence* a *iq*, jež každý má svůj jednoznačný význam.

Message

XML element prvního zanoření slouží k posílání zpráv všeho druhu. Je to základní metoda pro rychlý přenos informací z místa na místo. Zprávy jsou typu „push“, tedy jsou odeslány a neočekává se žádná aktivita od příjemce, která by přijetí potvrdila. Zprávy jsou používány pro IM, skupinový chat a pro oznámení nebo upozornění. Podelementy a atributy tvořící nutné minimum zprávy jsou *to* (příjemce zprávy), *from* (odesílatel zprávy) a *body* (obsah zprávy). Základní používané typy zpráv jsou *normální* (zpráva bez kontextu, vyžaduje odpověď), *chat* (komunikace mezi dvěma entitami), *groupchat* (skupinový chat) *headline* (upozornění) *error* (chybová zpráva). Celá struktura elementu `<message/>` je zobrazena v příloze E.2 strana 30.

Základní použití elementu `<message/>` je ukázáno na obrázku 2.4. Zobrazuje strukturu zprávy. Na prvním řádku je element *to* – příjemce zprávy, druhý *from* – odesílatel, následuje *type* – typ zprávy a nakonec samotný obsah.

```

1.      <message from="uzivatel@jabbim.com"
2.              to="jabinfo@jabbim.com/bot"
3.              type="chat"
4.      <body> Kolik je hodin? </body>
5.      </message>

```

Obrázek 2.4: Příklad použití elementu `<message/>`.

IQ

IQ nebo-li *Info/Query* poskytuje strukturu pro *request-response* (žádost–odpověď) vazbu a workflows, podobný metodám GET, POST a PUT z protokolu HTTP. Na rozdíl od elementu *message* tvoří *iq* spolehlivější přenos, optimalizovaný pro výměnu dat (binární data). Příjemce musí na každou přijatou zprávu odpovědět, nebo-li potvrdit přijetí. Žádosti na proces nebo akci jsou posílány jednotlivě [?]. Celá struktura elementu `<iq/>` je zobrazena v příloze E.1 strana 29.

Obrázek 2.5 znázorňuje základní použití elementu `<iq/>`. Uživatel *uzivatel* posílá dotaz na získání kontakt listu (řádek 5.).

```

1.      <iq from="uzivatel@jabbim.com/doma"
2.          to="uzivatel@jabbim.com"
3.          id="uhhfw23648"
4.          type="get"
5.      <query xmlns="jabber:iq:roster"/>
6.      </iq>

```

Obrázek 2.5: Příklad použití elementu `<iq/>`.

Presence

Presence nebo-li informace o stavu (přítomnost) rozesílá dostupnost ostatních entit v síti. Struktura elementu `<presence/>` je zobrazena v příloze E.3 strana 30.

Základní použití `<presence/>` je zobrazeno na obrázku 2.6. Kontakt *jabinfo@jabbim.com/bot* (1.řádek) posílá informace o svém stavu (řádek č.2) a svůj status (č.3).

```

1.      <presence from="jabinfo@jabbim.com/bot"
2.          <show> online </show>
3.          <status> Jsme zde. </status>
4.      </presence>

```

Obrázek 2.6: Příklad použití elementu `<presence/>`.

Jabber ID

Jabber ID (JID) je jednoznačný virtuální identifikátor uživatele na síti. Není case-sensitive a je složen ze dvou částí. Takzvané *Jabber bare* neboli čisté ID a *resource*. Základní část na

první pohled připomíná e-mailovou adresu *user@server*. Druhá část slouží k přesné identifikaci jednotlivých spojení. Je použita ke směrování trafiku s uživateli v případě otevření vícero spojení pod jedním uživatelem. Společně Jabber bare a resource tvoří tzv. *full JID* — *user@server/resource* například:

jabInfo@jabbim.cz/bot

2.4 Rozšíření

XMPP protokol je možné rozšířit o další vlastnosti k jejichž popisu slouží XEP. Pro tuto práci jsou nepostradatelné „statusy“, které popisují standardy XEP-0060 a XEP-0163 zkráceně PEP⁶. Obě tato rozšíření umožňují strukturovaně pracovat, používat a přenášet další XEP protokoly. Jako příklad jsou zde uvedeny protokoly XEP-0080 (User Location – co právě uživatel dělá) [6], XEP-0118 (User Tune – co uživatel poslouchá za hudbu) [15] a další, které jsou zmíněny v implementační části[...]. Jsou to tedy protokoly založené na PEP, které vyžadují podporu nejen v klientech, ale i na straně serveru (zobrazuje tabulka 2.2). S touto informací úzce souvisí další protokol XEP-0115 [7], který umožňuje zjistit podporované schopnosti klienta, případně které informace je ochoten přijímat viz 2.4.

Všechna tato rozšíření by mohla být přidána přímo do statusu viz 2.6, avšak ten je určen k informování o přítomnosti na IM síti. Hlavní rozdíl mezi PEP a obyčejným posílání stavu pomocí presence je v pravomoci klienta přijmout nebo odmítnout informaci, na rozdíl od presence, jež je přijata vždy.

Základ přenosu informací začíná na straně klienta, jež chce všechny ve svém roster (kontakt) listu informovat. Zašle zprávu zabalenou v elementu *<iq>* serveru. Příklad této zprávy je ukázán na obrázku 2.7, který znázorňuje zaslání informací o druhu hudby, kterou v danou chvíli poslouchá. Využívá k tomu rozšíření User Tune, definovaném na řádku číslo 5. Základ zprávy oznamující začátek vysílání informací o rozšířených statusech je vždy stejný. Liší se pouze řádkem 3. a obsahem elementu *item* v obrázku 2.7.

V případě úspěšného přijetí *<iq>* zprávy serverem, každý, kdo se zajímá o vaše rozšířené statusy, obdrží oznámení ve formě *<message>*. Část zprávy je vyobrazená na obrázku 2.8. Oznámení také obdrží všechny vaše resources. Celá zpráva i všechny další náležitosti jsou uvedeny v příloze E.4.

XEP-0115: Schopnosti subjektu

Pomocí protokolu Entity Capabilities je možné zjistit vlastnosti klienta. Tato informace výrazně snižuje počet a velikost komunikací a přenosů zpráv mezi uživateli. Dotazem zobrazeným na obrázku 2.9 je zjištěna schopnost jednotlivých klientů, kterou následně server využije pro správné směrování rozšířených statusů. Všechny zde zmiňované rozšíření a protokoly z kapitoly 2.4 je možné u každého klienta (seznam klientů obsahuje tabulka v příloze E.4) vyčíst z atributu *ver* (druhá část u atributu *node*), který je vypočítán ze všech podporovaných protokolů klienta, viz [7].

⁶Personal Eventing via Pubsub

```

1. <iq from='uzivatel@jabberim.com' type='set' id='pub1'>
2.   <pubsub xmlns='http://jabber.org/protocol/pubsub'>
3.     <publish node='http://jabber.org/protocol/tune'>
4.       <item>
5.         <tune xmlns='http://jabber.org/protocol/tune'>
6.           <artist>Daniel Landa</artist>
7.           <length>255</length>
8.           <source>Nigredo</source>
9.           <title>1968</title>
10          <track>5</track>
11         </tune>
12       </item>
13     </publish>
14   </pubsub>
15 </iq>

```

Obrázek 2.7: Příklad začátku vysílání rozšířeného statusu.

```

1. <message from='uzivatel@jabberim.com' type='set'
2.   to='jabinfo@jabberim.com/bot' id='pub1'>
3.   <event xmlns='http://jabber.org/protocol/pubsub#event'>
4.     <items node='http://jabber.org/protocol/tune'>
5.       <item>
6.         <tune xmlns='http://jabber.org/protocol/tune'>
7.           <artist>Daniel Landa</artist>
8.           <length>255</length>
9.           ...

```

Obrázek 2.8: Příklad 2. Zúčastněné strany obdrží upozornění na události.

2.5 Knihovny

Jabber je realizován jako otevřený XML standart pro instant messaging formát, proto existuje mnoho programovacích jazyků. Většina z nich disponuje několika knihovnami, usnadňující práci s protokolem XMPP. V tabulce 2.3 jsou pro nejznámější programovací jazyky zobrazeny dostupné knihovny.

gloox

Gloox je stabilní Jabber/XMPP knihovna vydávána pod licencí GNU GPL. Je určena pro vývoj klienta a komponent. Jelikož je psána v ANSCI C++ mezi podporované platformy patří Linux, Windows, Mac OS X, Symbian/Nokia S60, FreeBSD a další systémy stavějící na ANSI C++ kompilátor.

Pomocí knihovny gloox je psán bot v této práci. Byla vybrána na základě požadavku psaní programu v jazyce C/C++ a operačním systémem Linux. V porovnání s jinými knihovnami pro jazyk C nebo C++ disponuje lepší podporou a dokumentací. Gloox plně podporuje standart XMPP Core [17] a z větší části i standart XMPP IM [18]. Dodatečně je plně podporováno kolem 30 XEP standardů například XEP-0054 vcard-temp a další.

```

1. <iq from='uzivatel@jabbim.com' id='disco1'
2.   to='jabinfo@jabbim.com/bot' type='get'>
3.   <query xmlns='http://jabber.org/protocol/disco#info'
4.     node='http://code.google.com/p/exodus#QgayPKawpkPSDYmwT/WM94uAlu0=' />
5. </iq>

```

Obrázek 2.9: Dotaz na podporované protokoly.

Programovací jazyk	knihovna
C	iksemel, libstrophe, Loudmoutn
C++	gloox, Iris
JAVA	JabberBeans, Smack, JSO, Feridian, Emite, minijingle
.NET	Jabber-Net, agsXMPP SDK
Python	JabberPy, PyXMPP, SleekXMPP, Twisted Words
Perl	Net-Jabber
Ruby	XMPP4R, Jabber4R, Jabber::Simple, Jabber::Bot

Tabulka 2.3: Přehled Jabber knihoven

Kapitola 3

Data mining

...

Historycký vývoj

Pojem data mining nebo-li česky dolování dat se začal ve vědeckých kruzích objevovat počátkem 90. let 20. století. První zmínka pochází z konferencí věnovaných umělé inteligenci (IJCAI'89 ¹—mezinárodní konference konaná v Detroitu, AAAI'91 ² a AAAI'93—americké konference v Californii a Washingtonu, D.C).

Tradiční metoda získání informací z dat je realizována jejich manuální analýzou a interpretací. V praxi ji například nalezneme v odvětví zdravotnictví, vědy, marketingu (efektivita reklamních kampaní, segmentace zákazníků) a dalších. Pro tato a mnoho dalších disciplín je manuální zpracování příliš pomalé, drahé a vysoce subjektivní. Na druhou stranu velikost dat dramaticky vzrostla tudíž se manuální analýza stává zcela nepraktická. Databáze rychle rostou ve dvou kategoriích :

1. počet záznamů nebo-li objektů v databázi
2. počet polí nebo-li atributů objektů v databázi

Proces data mining je pouze jedna část z odvětví nazývané dobývání znalostí z databází nebo-li KDD (dále jen zkratka KDD) ³ definované níže 3.0.1. Vznik disciplíny KDD je důsledkem nepřehledného množství automaticky sbíraných dat, která je potřeba dále využívat. Podstatným znakem celého procesu je správnost reprezentace výsledků formou uživateli nejbližší. Jako příklad uveďme implikace ve tvaru rozhodovacích pravidel, asociační pravidla, rozhodovací stromy, shluky podobných dat a další. Základem KDD je praktická použitelnost metod. Očekává se zjištění nových skutečností namísto prezentování již známých informací.

Definice 3.0.1 KDD je chápáno jako interaktivní a iterativní proces tvořený kroky selekce, předzpracování, transformace, vlastního „dolování“ (data-mining viz 3.0.2) a interpretace [2].

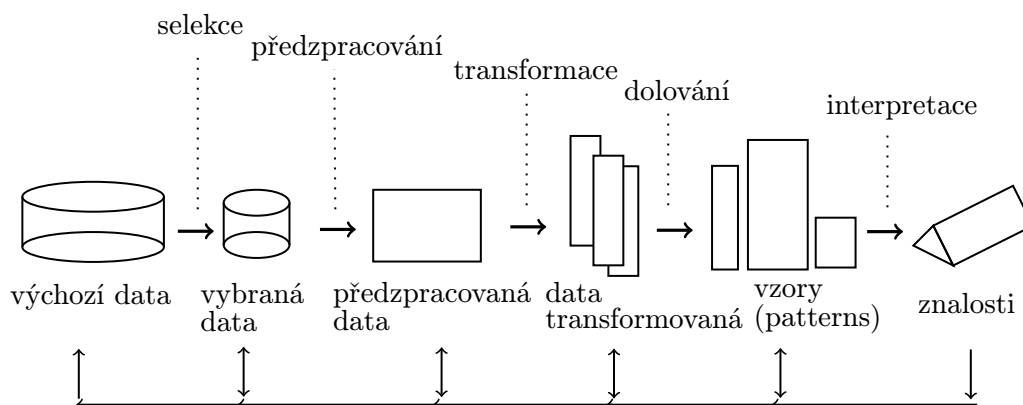
Grafické znázornění definice 3.0.1 je popsáno schématem na obrázku 3.1, který ukazuje časový harmonogram v KDD. Schéma znázorňuje následnost jednotlivých procesů, které

¹International Joint Conference on Artificial Intelligence

²Association for the Advancement of Artificial Intelligence

³Knowledge Discovery in Database

tvorí KDD. KDD je iterativní proces. Skutečnosti nalezené v předešlých částí zjednodušují a zpřesňují vstupy pro následující fáze. Jakmile jsou znalosti získány, jsou předvedeny uživateli. Pro přesnost může být část procesu KDD ještě upravena. Tím budou získány „přesnější a vhodnější“ výsledky.



Obrázek 3.1: Proces dobývání znalostí z databází podle knihy autora Fayyad [3].

Získávání znalostí z databází je proces složený z několika kroků vedoucích od surových dat k formě nových poznatků. Iterativní proces se skládá z následujících kroků:

- **čištění dat** – fáze, ve které jsou nepodstatné údaje odstraněny z kolekce.
- **integrace dat** – kombinování heterogenních dat z několika zdrojů do společného jediného zdroje.
- **výběr dat** – rozhodování o relevantních datech.
- **transformace dat** – také známý jako konsolidace dat. Fáze, ve které jsou vybraná data transformována do formy vhodné pro dolování.
- **data mining** – zásadní krok, ve kterém jsou aplikovány vzory na data.
- **hodnocení modelů** – vzory dat zastupují získané znalosti.
- **prezentace znalostí** – konečná fáze, zjištěné poznatky jsou reprezentovány uživateli. Tento základní krok využívá vizualizační techniky, které pomáhají uživatelům porozumět a správně interpretovat získané výsledky.

Je běžné některé z těchto kroků kombinovat dohromady. Krok čištění dat a integrace dat může být provedena společně, tak jako je uvedeno na obrázku 3.1.

Základní pojmy

V této podsekcí jsou ve stručnosti vysvětleny základní nejdůležitější pojmy dále v práci využívané.

Definice výrazu data mining se v odborné literatuře nachází několik. Zde uvedená je kombinací dvou „definic“. **Za předchůdce tohoto oboru se považuje vědní obor statistika, ze které se postupem času vyčlenil.**

Definice 3.0.2 Data Mining je proces objevování znalostí, který používá různé analytické nástroje sloužící k odhalení dříve neznámých vztahů a informací z velmi rozsáhlých databází. Výsledkem je predikční model, který je podkladem pro rozhodování [14].

Mezi další čteně se vyskytující pojmy v tomto odvětví patří například data, znalosti a informace. Tyto termíny jsou často mezi sebou zaměňovány, proto je níže jejich význam striktně definován.

Jedna z několika existujících definic pojmu data je uvedena v 3.0.3, která je popisuje z pohledu informačního. Data často nemají sémantiku (význam) a bývají zpracována čistě formálně.

Definice 3.0.3 Data jsou z hlediska počítačového pouze hodnoty různých datových typů [8].

Informace definované v 3.0.4, jsou zpracovaná data interpretována uživatelem.

Definice 3.0.4 Informace jsou data, která mají sémantiku (význam) [8].

Znalosti, definované v 3.0.5, patří do stejné kategorie jako informace, ale jejich interpretace bývá ještě složitější. Často jsou tvořeny shluky informací, proto bývají reprezentovány jako odvozené informace.

Definice 3.0.5 Znalosti jsou informace po jejich zařazení do souvislostí [8].

3.1 Metody dolování dat

Základ metod dolování dat je založen na statistice, posledních poznatcích z umělé inteligence či strojového učení. Hlavní cíle těchto netriviálních metod je společný – snaha zjištěné výsledky prezentovat srozumitelnou formou. Pro většinu používaných metod je společná vlastnost předpoklad, že objekty popsané pomocí podobných charakteristik patří do stejné skupiny (učení na základě podobnosti similarity-based learning). Objekty obsahující atributy, lze převést na body v n -rozměrném prostoru (n – počet atributů). Vychází z představy podobnosti bodů tvořící určité shluky v prostoru.

Tyto dva příklady nejsou jediné rozdíly v metodách dolování dat. Další rozdíly spočívají v:

- schopnost reprezentace shluků (např. otázka lineární separability)
- srozumitelnost nalezených znalostí pro uživatele (symbolické vs. subsymbolické metody)
- efektivnost znovupoužití nalezených znalostí
- vhodnost typů data
- a další ...

Problémy, které data mining řeší, se rozdělují do několika skupin. Výčet vybraných z nich nejčastějších:

Asociační pravidla

Asociační pravidla jsou založena na syntaxi IF-THEN. Jejich rozšíření se datuje do 90.let 20.století. Pan Agrawal je představil v souvislosti s analýzou „nákupního košíku“.

Použitelnost bude vysvětlena právě na příkladu analýzy nákupního košíku. Podstata příkladu je tvořena zákazníkem a jeho systémem nakupování. Jsou zjišťovány produkty, které jsou nakupovány současně. Hledáme nebo-li vytváříme společné vazby (asociační pravidla) mezi výrobky a určujeme jejich spolehlivost. Na základě těchto závislostí se upravuje umístění jednotlivých výrobků.

Obecně lze tedy asociační pravidla považovat za konstrukci, která z hodnot jedné transakce odvozuje možnost výskytu závislostí v jiných transakcích. Jsou hledány všechny vnitřní závislosti mezi daty.

Z pravidel vytvořených z dat zjišťujeme počet příkladů splňujících předpoklad a kolik závěr pravidla, kolik příkladů splňuje obě podmínky a počet příkladů splňující pouze druhou část pravidla. Podle knihy Berky [2] převedeme základní myšlenku asociačních pravidel *IF-THEN* do jiné terminologie:

$$\text{Ant} \Rightarrow \text{Suc}$$

kde *Ant* bývá interpretován jednou z možností výčtu – předpoklad, IF, levá strana pravidla nebo antecedent a *Suc* je chápán jako – závěr, ELSE, pravá strana pravidla, sukcedent. Nyní jsou uvedeny základní vlastnosti:

$$n(\text{Ant} \wedge \text{Suc}) = \mathbf{a}; n(\text{Ant} \wedge \neg \text{Suc}) = \mathbf{b}; n(\neg \text{Ant} \wedge \text{Suc}) = \mathbf{c}; n(\neg \text{Ant} \wedge \neg \text{Suc}) = \mathbf{d}; \\ n(\text{Ant}) = a+b = \mathbf{r}; n(\neg \text{Ant}) = c+d = \mathbf{s}; n(\text{Suc}) = a+c = \mathbf{k}; n(\neg \text{Suc}) = b+d = \mathbf{l}; n = a+b+c+d;$$

všechna pravidla jsou shrnuta v tabulce 3.1, z nichž jsou dále počítány různé charakteristiky a následně tak hodnotit zjištěné znalosti.

	Suc	¬Suc	Σ
Ant	a	b	r
¬Ant	c	d	s
Σ	k	l	n

Tabulka 3.1: Kontingenční tabulka [2].

Mezi základní charakteristiky asociačních pravidel podle Agrawalova patří *podpora* a *spolehlivost*.

Klasifikace

Klasifikace bude opět vysvětlena na příkladu. Obsah databáze nebo dotazníku nám každého klienta banky zařadí do různých krizových skupin. Na základě těchto skupin pracuje „credit skóring“, jež klientovi poskytne nebo odepře například úvěr v bance. Dalším příkladem je zdravotnictví. Na základě zdravotního stavu pacienta a příznaků je zařazen do tříd, reprezentujících jednotlivé nemoci.

Klasifikace rozdělí jednotlivé zkoumané elementy (podle hodnot atributů) do vhodných kategorií, které jsou předem vytvořeny navzájem podobnými objekty (tvorba profilů třídy). Při této metodě se upřednostňuje přesnost před jednoduchostí a rychlostí. Je snaha o nalezení obecných závěrů z chování dat. Zdroje klasifikovaných objektů většinou tvoří jednotlivé

řádky v databázi. Vzory dat vytváří instance. Jejich vlastnosti reprezentují atributy vyjádřené číselnou hodnotou.

Z matematického pohledu klasifikace představuje funkci o více proměnných. Atribut instance odpovídá proměnné a funkční hodnota výstupu.

Modely

Základem modelů jsou trénovací data. Příklad vybraných klasifikačních modelů:

- Rozhodovací stromy
- Neuronové sítě
- Statistické metody
- Klasifikační pravidla
- Využití vzdálenosti
- a další.

Predikce

Predikce patří mezi velmi známé procesy, které na základě získaných znalostí předpovídají následující vývoj. Chronologicky seřazená data a vývoj jejich hodnot v minulosti tvoří základ pro určení hodnot budoucích. Předpokládá se, že data v minulosti budou chovat stejně nebo alespoň podobně i v budoucnu. Využití naleznou v předpovědi počasí (z naměřených meteorologických hodnot se určují budoucí předpokládané teploty), při vývoji cen na burze a dalších.

Shlukování

Metoda zaměřená na dělení objektů do předem neznámých skupin. Proces dělení probíhá na základě specifikace objektů a jejich odlišnosti od ostatních shluků. Tato část bude podrobně rozebrána v následující kapitole 3.2.

3.2 Shlukování

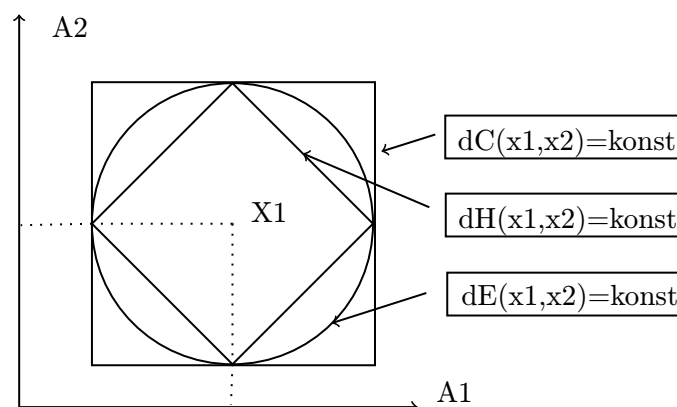
V této podkapitole je shlukování rozděleno na několik metod shlukové analýzy podle [5]. U každé z nich jsou popsány její základní vlastnosti a uvedeny nejrozšířenější algoritmy. Poslední metoda *metoda rozkladu* 3.2 je rozebrána podrobněji z důvodu jejího praktického využití v této práci.

Shlukování je zaměřeno na dělení objektů do předem neznámých skupin. Proces dělení probíhá na základě specifikace objektů a jejich odlišnosti od ostatních shluků.

Výpočet vzdáleností mezi objekty patří, pro většinu níže uvedených metod a algoritmů, mezi nedílnou součást. Tato vzdálenost lze vyjádřit různými mírami například *Hammingovou vzdáleností* (dH), *Euklidovskou vzdáleností* (dE) a *Čebyševovou vzdáleností* (dC). Rozdíl mezi těmito typy určující vzdálenosti, graficky vyjadřuje obrázek č. 3.2.

Metody hierarchické

...



Obrázek 3.2: Srovnání výpočtu vzdáleností od bodu x_1 [2].

Metody založené na mřížce

Metody založené na mřížce kvantují datový prostor do konečného počtu pravoúhlých buněk, které jsou uspořádány do víceúrovňové mřížkové struktury. Zmíněná struktura tvoří základ pro shlukové operace. Hlavní výhodou tohoto přístupu je rychlost zpracování, které většinou nebere ohled na počet datových objektů. Čas zpracování závisí pouze na počtu buněk v každé dimenzi kvantovaného prostoru.

Mezi zástupce metod založených na mřížce patří metoda STING⁴, který pracuje se statickými informacemi uloženými v buňkách mřížky. Algoritmus je rozdělen do dvou částí. První si klade za cíl rekurzivně rozdělit datový prostor na pravoúhlé buňky. Druhá fáze testuje spojitost mezi sousedy relevantních buněk.

Mezi další metody založené na mřížce patří CLIQUE⁵ a WaveCluster⁶. WaveCluster využívá vlnkové transformace k rozdělení prostoru dat. Tato transformace zdůrazňuje shluky v prostoru a objekty jim vzdálené potlačuje.

Metody založené na modelu

Metody založené na modelu se pokouší přiřadit data k určitému matematickému modelu na základě společných optimalizovaných vlastností. Většina procesů je založena na předpokladu generování dat pomocí standardních statistik.

Mezi zástupné metody této shlukovací analýzy se řadí Expectation–Maximization (EM), SOON⁷ a částicové filtry (particle filters). Algoritmus SOON je založen na neuronové síti. Je to metoda vycházející z algoritmu SOM⁸. Metoda EM je rozšířením algoritmu k -means o němž se budeme podrobně zabývat v následujících částí 3.2.

Metody založené na hustotě

Vychází z m -rozměrného prostoru do něhož, jsou zobrazeny objekty ve formě bodů. Místa v prostoru s větší koncentrací objektů ve srovnání s ostatními oblastmi jsou nazývány

⁴Statistical INformation Grid

⁵CLustering In QUEst

⁶Clustering Using Wavelet Transformation

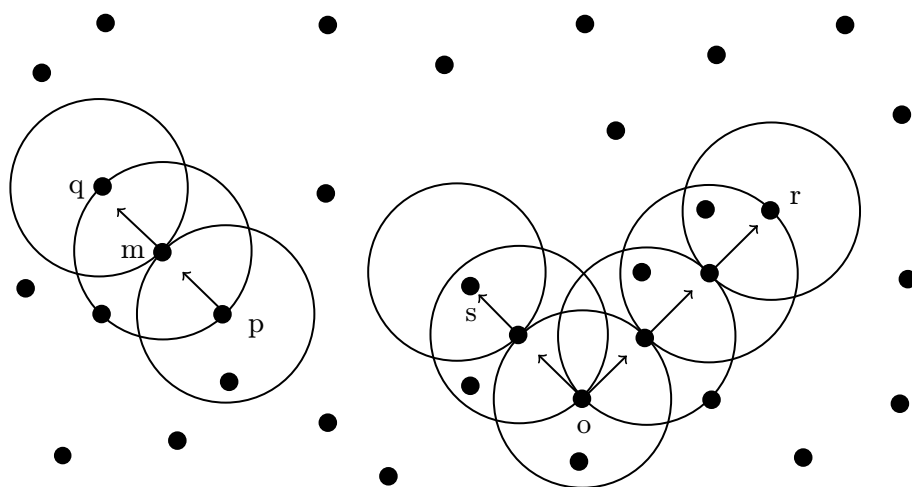
⁷Self Organizing Oscillator Network

⁸Self-Organizing Map

shluky. Jsou to ta místa, která hledáme. Výchozí předpoklad je existence okolí jednotlivých bodů (sousedství). Tato metoda se dokáže vypořádat s hodnotami velmi vzdálenými označovanými jako šum.

Jako příklad uveďme metody DBSCAN⁹, OPTICS¹⁰ a DENCLUE¹¹. Metoda DBSCAN je založena na hustotě objektů v prostoru. U jednotlivých objektů je zkoumáno jejich okolí. Algoritmus je ovlivňován dvěma parametry ε (velikost shluku) a $MinPts$ (minimální počet objektů v daném shluku), které spolu úzce souvisí (viz [5]). Bod splňující obě podmínky je označen za jádro. Pomocí jader se rozšiřuje množina objektů spojených na základě hustoty. Obsahuje-li jádro x_1 ve svém okolí další jádro x_2 znamená to, že jádro x_1 je přímo dosažitelné z jádra x_2 . Tímto způsobem se vytváří výsledné *shluky*. V opačném případě, body nesplňující dvě zmíněné podmínky, jsou označeny jako *šum*.

Příklad postupu algoritmu DBSCAN zobrazuje obrázek č. 3.3. Kde ε -sousedství tvoří kruh a $MinPts$ má hodnotu 3.



Obrázek 3.3: Algoritmus DBSCAN [5].

Shlukování velkých dat

Všechny zde doposud zmiňované metody poskytují dobré výsledky pouze s malým počtem dimenzí. S narůstajícím počtem atributů roste počet nerelevantních dimenzí určených pro shlukování. S tímto také přibývá zvětšená produkce zašumění a znesnadnění nalezení relevantních shluků. Data jsou roztroušena do mnoho dimenzí a tím odpadá možnost použití vzdálenostních funkcí.

Zmíněné problémy shlukování velkých dat řeší dvě techniky *metoda transformace rysů* a *metoda výběru atributů*.

Pro efektivní shlukování je možné použít například algoritmus CLIQUE¹², PROCLUS¹³, ENCLUS¹⁴ a další.

⁹Density-Based Spatial Clustering of Applications with Noise

¹⁰Ordering Points To Identify the Clustering Struktüre

¹¹DENsity-base CLUstering

¹²CLustering In QUEst

¹³PROjested CLUstering

¹⁴ENntropy-based CLUstering

Metody rozkladu

...

k-Means

...popis a za pis algoritmu

k-Medoids

...

3.3 Programy

Fitminer

Rapid miner

Weka

Kapitola 4

Developer

4.1 Databáze

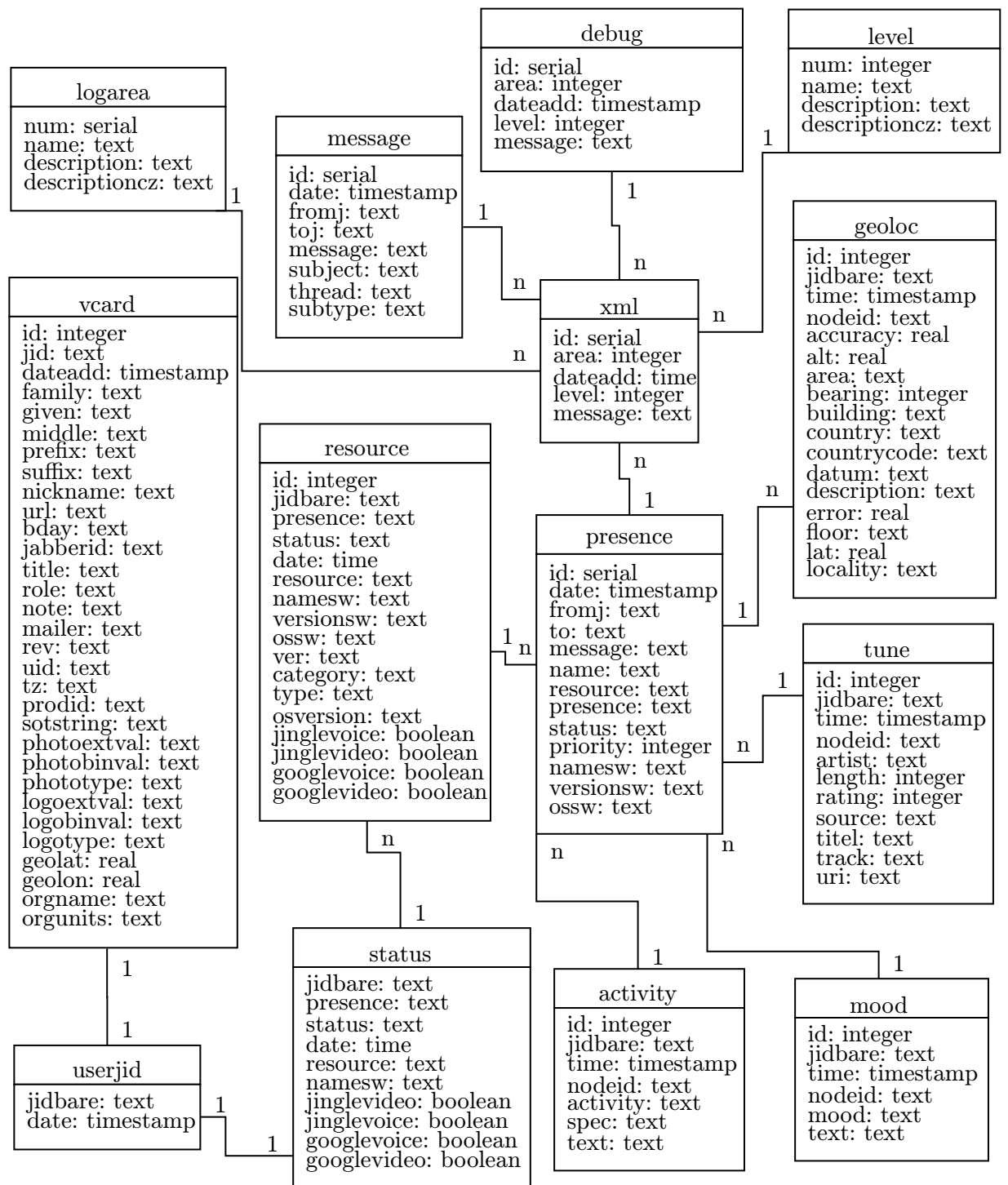
Návrh databáze

4.2 Bot

Návrh bota

4.3 Knihovny, jazyk

4.4 Jiné produkty



Obrázek 4.1: Struktura databáze

Kapitola 5

Vyhodnocení výsledků

Kapitola 6

Závěr

Literatura

- [1] Adams, D.: *Programming jabber*. Sebastopol: O'Reilly, první vydání, 2002, 455 s., iISBN 05-960-0202-5.
- [2] Berka, P.: *Dobývání znalostí z databází*. Praha: Academia, první vydání, 2003, 366 s., iISBN 80-200-1062-9.
- [3] Fayyad, U. M.; Smyth, P.: *Advances in knowledge discovery and data mining*. California: MIT Press, první vydání, 1996, 611 s., iISBN 02-625-6097-6.
- [4] Fred, H.: *Computer networking and the internet*. Edinburg: Addison-Wesley Publishing Company, první vydání, 2005, 803 s., iISBN 03-212-6358-8.
- [5] Han, J.; Kamber, M.: *Data mining : concepts and techniques*. San Francisco: Morgan Kaufmann Publisher, druhé vydání, 2006, 770 s., iISBN 15-586-0901-6.
- [6] Hildebrand, J.; Saint-Andre, P.: XEP-0080: User Location. [online], 15-09-2009, [cit. 14. dubna 2011].
URL <http://xmpp.org/extensions/xep-0080.html>
- [7] Hildebrand, J.; Saint-Andre, P.; Tronçon, R.; aj.: XEP-0115: Entity Capabilities. [online], 02-26-2008, [cit. 14. dubna 2011].
URL <http://xmpp.org/extensions/xep-0115.html>
- [8] Hruška, T.: *Informační systémy : IIS/PIS*. Brno: Fakulta informačních technologií, 2008, 14733 s.
- [9] Kolektiv autorů: Extensible Markup Language (XML) 1.0. [online], 26-11-2008, [cit. 14. dubna 2011].
URL <http://www.w3.org/TR/2008/REC-xml-20081126/>
- [10] Kosek, J.: *XML pro každého : podrobný průvodce*. Praha: Grada, první vydání, 2000, 163 s., iISBN 80-716-9860-1.
- [11] Kurose, J. F.; Ross, K. W.: *Computer networking : top-down approach featuring the internet*. Boston: Addison-Wesley Publishing Company, druhé vydání, 2003, 752 s., iISBN 03-211-7644-8.
- [12] Millard, P.; Saint-Andre, P.; Meijer, R.: XEP-0060: Publish-Subscribe. [online], 12-07-2010, [cit. 14. dubna 2011].
URL <http://xmpp.org/extensions/xep-0060.html>
- [13] Moore, D.; Wright, W.: *Jabber developer's handbook*. Indianapolis: Sams Publishing, první vydání, 2004, 487 s., iISBN 06-723-2536-5.

- [14] Nemrava, M.; Pospíšil, J.: Dolování dat a jeho aplikace. [online], 2006, [cit. 14. dubna 2011].
URL http://www.spatial.cs.umn.edu/paper_ps/dmchap.pdf
- [15] Saint-Andre, P.: XEP-0118: User Tune. [online], 30-01-2008, [cit. 14. dubna 2011].
URL <http://xmpp.org/extensions/xep-0118.html>
- [16] Saint-Andre, P.; Smith, K.: XEP-0163: Personal Eventing Protocol. [online], 12-07-2010, [cit. 14. dubna 2011].
URL <http://xmpp.org/extensions/xep-0163.html>
- [17] Saint-André, P.: Extensible Messaging and Presence Protocol (XMPP): Core. [online], 10-2004, [cit. 14. dubna 2011].
URL <http://tools.ietf.org/html/rfc3920>
- [18] Saint-André, P.: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence. [online], 10-2004, [cit. 14. dubna 2011].
URL <http://tools.ietf.org/html/rfc3921>
- [19] Saint-André, P.; Smith, K.; Troncon, R.: *XMPP : the definitive guide : building real-time applications with jabber technologies*. Sebastopol: O'Reilly, první vydání, 2009, 287 s., iISBN 978-059-6521-264.
- [20] Stevens, W.; Fenner, B.; M.Rudoff, A.: *UNIX Network Programming*. Boston: Addison-Wesley Publishing Company, třetí vydání, 2004, 991 s., iISBN 01-314-1155-1.

Příloha A

Obsah CD

Příloha B

Manual

Příloha C

Konfigurační soubor

Příloha D

Slovník výrazů

DNS — Domain Name System

GPG — dkshckdsjvlsdjvodsvjdfokj

IM služby — dkshckdsjvlsdjvodsvjdfokj

IP — Internet Protocol

JEP — dkshckdsjvlsdjvodsvjdfokj

JID — dkshckdsjvlsdjvodsvjdfokj

SASL — dkshckdsjvlsdjvodsvjdfokj

TCP — dkshckdsjvlsdjvodsvjdfokj

TLS — dkshckdsjvlsdjvodsvjdfokj

WWW — dkshckdsjvlsdjvodsvjdfokj

XEP — dkshckdsjvlsdjvodsvjdfokj

XML — dkshckdsjvlsdjvodsvjdfokj

XMPP — dkshckdsjvlsdjvodsvjdfokj

e-mail — dkshckdsjvlsdjvodsvjdfokj

jabber — dkshckdsjvlsdjvodsvjdfokj

klient — dkshckdsjvlsdjvodsvjdfokj

presence — dkshckdsjvlsdjvodsvjdfokj

server — dkshckdsjvlsdjvodsvjdfokj

stanza — dkshckdsjvlsdjvodsvjdfokj

vCard — dkshckdsjvlsdjvodsvjdfokj

Příloha E

Stanza - základní schéma

E.1 iq

```
<iq from=""  
  to=""  
  type="[get,set,result,error]"  
  id=""  
  Namespace  
</iq>
```

Obrázek E.1: Popis elementu *<iq/>*.

jabber:client	jabber:server	jabber:iq:auth	jabber:iq:register
jabber:iq:roster	jabber:x:offline	jabber:iq:agent	jabber:iq:agents
jabber:x:delay	jabber:iq:version	jabber:iq:time	vcard-temp
jabber:iq:private	jabber:iq:search	jabber:iq:oob	jabber:x:oob
jabber:iq:admin	jabber:iq:filter	jabber:iq:auth:0k	jabber:iq:browse
jabber:x:event	jabber:iq:conference	jabber:x:signed	jabber:x:encrypted
jabber:iq:gateway	jabber:iq:last	jabber:x:envelope	jabber:x:expire
jabber:xdb:ginsert	jabber:xdb:nslist	texthttp://www.w3.org/1999/xhtml	

Tabulka E.1: Přehled Namespace elementu *<iq/>*.

E.2 Message

```
<message from=""
  to=""
  type="[normal,chat,groupchat, headline, error]"
  id=""
  <body> </body>
  <x xmlns="jabber:x:event">
    <[Offline, Delivered, Displayed, Composing]/>
  </x>
  <subject> </subject>
  <thread> </thread>
  <error> </error>
</message>
```

Obrázek E.2: Popis elementu *<message/>*.

E.3 Presence

```
<presence from=""
  to=""
  type="[available, unavailable, probe, subscribe,
        unsubscribe, subscribed, unsubscribed, error]"
  id=""
  <show>
    [away, chat, dnd, normal, xa]
  </show>
  <status> </status>
  <priority> </priority>
  <error> </error>
</presence>
```

Obrázek E.3: Popis elementu *<presence/>*.

E.4 Přehled průběhu rozšíření

```
<iq from="uzivatel@jabbim.com" type="set" id="pub1">
  <pubsub xmlns="http://jabber.org/protocol/pubsub">
    <publish node="http://jabber.org/protocol/tune">
      <item>
        <tune xmlns="http://jabber.org/protocol/tune">
          <artist>Daniel Landa</artist>
          <length>255</length>
          <source>Nigredo</source>
          <title>1968</title>
          <track>5</track>
        </tune>
      </item>
    </publish>
  </pubsub>
</iq>
```

Obrázek E.4: Informování server o právě přehrávající písničce.

```
<message from="uzivatel@jabbim.com" type="set"
  to="jabinfo@jabbim.com/bot" id="pub1">
  <event xmlns="http://jabber.org/protocol/pubsub#event">
    <items node="http://jabber.org/protocol/tune">
      <item>
        <tune xmlns="http://jabber.org/protocol/tune">
          <artist>Daniel Landa</artist>
          <length>255</length>
          <source>Nigredo</source>
          <title>1968</title>
          <track>5</track>
        </tune>
      </item>
    </items>
  </event>
</message>
```

Obrázek E.5: Server informuje uživatele podporující rozšíření o stavu *uzivatel@jabbim.com*.

```

<message from="uzivatel@jabber.com" type="set"
  to="uzivatel@jabber.com/doma" id="pub2">
  <event xmlns="http://jabber.org/protocol/pubsub#event">
    <items node="http://jabber.org/protocol/tune">
      <item>
        <tune xmlns="http://jabber.org/protocol/tune">
          <artist>Daniel Landa</artist>
          <length>255</length>
          <source>Nigredo</source>
          <title>1968</title>
          <track>5</track>
        </tune>
      </item>
    </items>
  </event>
</message>

```

Obrázek E.6: Server přepošle informace o přehrávané písničce všem otevřeným spojením uživatele *uzivatel@jabber.com*.

```

<iq from="uzivatel@jabber.com/prace" type="set" id="pub1">
  <pubsub xmlns="http://jabber.org/protocol/pubsub">
    <publish node="http://jabber.org/protocol/tune">
      <item>
        <tune xmlns="http://jabber.org/protocol/tune"/>
      </item>
    </publish>
  </pubsub>
</iq>

```

Obrázek E.7: Uživatel ukončil „vysílání“ rozšířených zpráv o svém stavu.

```

<message from="uzivatel@jabber.com"
  to="jabinfo@jabber.com/bot">
  <event xmlns="http://jabber.org/protocol/pubsub#event">
    <items node="http://jabber.org/protocol/tune">
      <item>
        <tune xmlns="http://jabber.org/protocol/tune"/>
      </item>
    </items>
  </event>
</message>

```

Obrázek E.8: Server informuje klienty o ukončení šíření rozšířeného statusu uživatele *uzivatel@jabber.com*.

Příloha F

Přehled klientů a jejich rozšíření

..... sdufiysdiufisudfyn df ..

. zrofj <http://xmpp.org/xmpp-software/clients/> ..sd.fsd

Klient	OS	XEP--60	XEP--163	XEP--80	XEP--92	XEP--107	XEP--108	XEP--118	XEP--	XEP--
Adium										
Agile Messenger										
AQQ										
Ayttm										
beejive										
Beem										
BitlBee										
Bombus										
BuddyMob										
Chatopus										
Citron										
Claros Chat										
climm										
Coccinella										
Crosstalk										
Digsby										
eM Client										
emite										
Empathy										
Exodus										
Finch										
Gajim										
Galaxium										
glu										
GNU Freetalk										
Gossip										
iChat										
iJab										
IM+										
imov Messenger										
irssi-xmpp										
Jabbear										
Jabber Mix Client										
jabber.el										
Jabbim										
Jabbim for Android										
Jabiru										
JAJC										
Jappix										

Klient	OS	XEP--60	XEP--163	XEP--80	XEP--92	XEP--107	XEP--108	XEP--118	XEP--	XEP--
JBuddy Messenger										
Jeti										
Jitsi (SIP Communicator)										
JWChat										
Kadu										
Kopete										
Lampiro										
m-im										
mcabber										
mChat										
Miranda IM										
Monal IM										
OctroTalk										
OneTeam										
OneTeam for iPhone										
Oyo										
Pandion										
Poezio										
Pidgin										
Prodromus										
Psi										
Psi+										
Quiet Internet Pager (QIP)										
qutIM										
saje										
SamePlace										
Sim-IM										
Slimster										
SoapBox Communicator										
Spark										
SparkWeb										
Synapse										
Talkonaut										
Tigase Messenger										
Tigase Minichat										
Tkabber										
Tlen										
Trillian										
TrophyIM										
V&V Messenger										
Vacuum-IM										
Vayusphere										
WTW										
Xabber										
xmppchat										
Yambi										
Yaxim										

Tabulka F.1: Přehled podporovaných rozšíření u jednotlivých klientů.