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

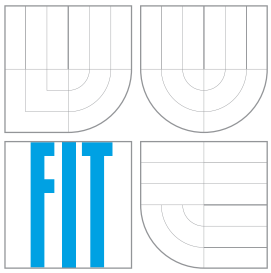
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

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 — JAROSLAV SENDLER
AUTHOR

VEDOUCÍ PRÁCE — Ing. JOZEF MLÍCH
SUPERVISOR

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
11. 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.

Obsah

Kapitola 1

Úvod

Dnes mezi velmi se rozšiřující technologie na poli síťo

V současné době se v marketingovém odvětví hledají jiné formy pro prezentaci zboží a služeb, jelikož zákazníci přehlíží klasické formy marketingu i marketingové komunikace. Proto se hledají a vyvíjí nové marketingové trendy, které zákazníky zaujmou, a v mnoha případech se jedná o takové formy, které mohou zvýšit účinnost propagace za nižší náklady. Cílem bakalářské práce je zjištění míry využívání nových trendů v marketingu v Obchodní pasáži Rozkvět. Tato práce zodpoví otázku, zda Rozkvět využívá nové trendy v marketingu, v jaké míře a jakým konkrétním způsobem.

Kapitola 2

XMPP

Pro usnadnění 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 [16, 17] a další detaily protokolu [1, 18, 12], které se vztahují k požadavkům na data mining popisovaný v této práci [15, 11]. Další informace byly čerpány z [19, 10, 4, 9, 8].

Samotný protokol je datován do roku 2004 (březen), kdy na něj byl přejmenován Jabber. Původní projekt Jabber byl vytvořen roku 1998 autorem Jeremie Miller, jež ho založil na popud nesvobodných uzavřených IM služeb. Měl mít tři základní vlastnosti -jednoduchost a srozumitelnost pro implementaci, jednoduše rozšiřitelný a otevřený. 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 napsala 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 se může 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-mail[1]. S každou zde jmenovanou službou má Jabber něco společného. Co se týče charakteristiky se vydal

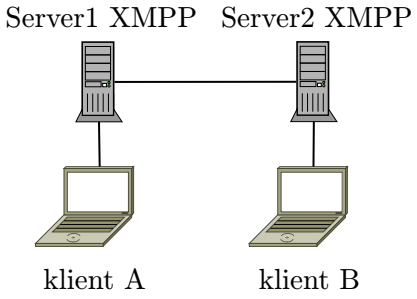
[1] internetový systém elektronické pošty

střední cestou. Na rozdíl od e-mailu, nepoužívá vícenásobný hops a v porovnáním 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"JID[2], 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 odesíláni zpráv, ale v této práci jej zastupuje bot 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 soketu
2. rozparsování a následná interpretace příchozí XML zprávy „stanza"(kapitola 2.3)
3. porozumění sadě zpráv z Jabber jádra

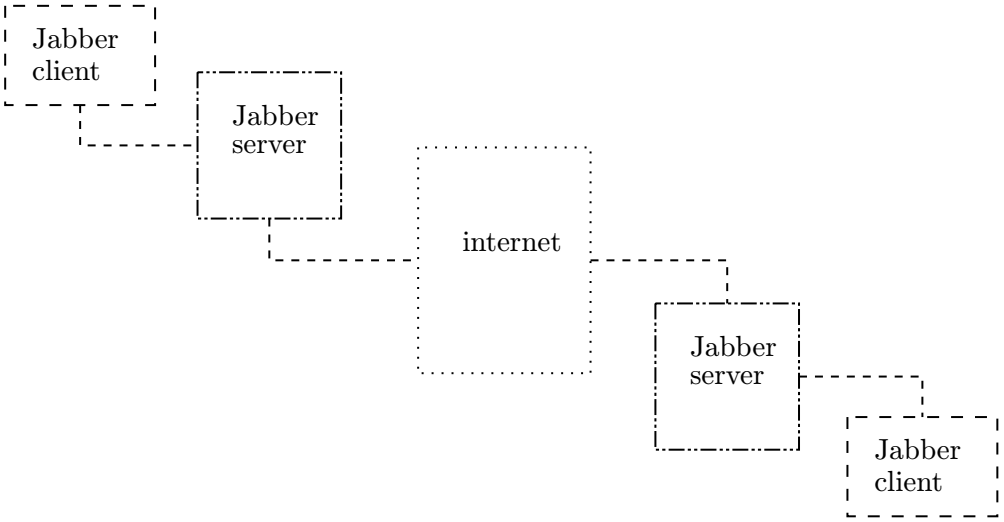
Server

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" účtu. Protože XMPP komunikace probíhá přes síť, musí mít každá entita adresu, tady nazvána JabberID. XMPP spoléhá na DNS tudíž používá jména na rozdíl od IP protokolu.

Obrázek 2.2 znázorňující distribuovanou architekturu Jabberu byl převzat z [12].

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 licencí GPL[3],

[2]uživatelské jméno
[3]General Public License-všeobecná veřejná licence GNU



Obrázek 2.2: Distribuovaná architektura Jabberu

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/ Top	2.1.6	ANO	ANO
Openfire	java	3.6.4	ANO	ANO
jabbred2	c	2.2.11	NE	NE
jabbred14	c, c++	1.6.1.1	ANO[4]	NE
Prosody	lua	0.7.0	NE [5]	ANO
Tigase	java	5.0.0	ANO	ANO

Tabulka 2.2: Přehled Jabber serverů

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 pro 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 kódovaní, 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 sloužící 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. Pod elementy 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 27.

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 wokrflows, podobný metodám GET, POST a PUT z protokolu HTTP. Na rozdíl od *message* je *iq* spolehlivější přenos optimalizovaný pro výměnu dat (binární data). Příjemce musí na každou přijatou zprávu odpovědět, neboli potvrdit přijetí. Žádosti na proces nebo akci jsou posílány jednotlivě [**?**]. Celá struktura elementu <*iq/*> je zobrazena v příloze E.1 strana 26.

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 27.

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*. První je část na první pohled připomíná e-mailovou adresu *user@server*. Druhá část slouží k přesné identifikaci jednotlivých spojení. Je použit 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. Pro jejich popis slouží XEP. Pro tuto práci jsou nepostradatelné „statusy“, které popisují standardy XEP–0060 a XEP–0163 zkráceně PEP [6]. 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) [5], XEP–0118 (User Tune – co uživatel poslouchá za muziku) [14] 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 [6], který umožňuje zjistit co jaký klient podporuje případně, které informace je ochoten přijímat viz. 2.4.

Jistě, že by se toto všechno mohlo být přidáno 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 odmítnou 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 <*iq*> serveru. Příklad této zprávy je ukázán na obrázku 2.7, který znázorňuje zaslání informaci o písničce, kterou zrovna poslouchám. 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.

```
1.  <iq from='uzivatel@jabbim.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.

V případě úspěšné 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.....

[6] Personal Eventing via Pubsub

```
1.  <message from='uzivatel@jabbim.com' type='set'
2.           to='jabinfo@jabbim.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.

XEP-0115: Schopnosti subjektu

Pomocí protokolu Entity Capabilities je možné zjistit vlastnosti klienta. Tato informace výrazně snižuje počet a velikost komunikace a přenos zpráv mezi uživateli. Dotazu zobrazený na obrázku 2.9 je zjištěna schopnost jednotlivých klientů a server tak následně ví, kam jednotlivě rozšířené statusy zasílat. Všechny zde zmiňované rozšíření a protokoly z kapitoly [...] je možné u každého klienta vyčíst z atributu *ver* (druhá část u atributu node), který je vypočítán ze všech podporovaných protokolů klienta, viz. [6].

```
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.

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.

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

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 podporující 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 systému Linux. V porovnání s jinými knihovnami pro jazyk C nebo C++ disponuje lepší podporovou a dokumentací. Gloox plně podporuje standart XMPP Core [16] a z větší části i standard XMPP IM [17]. Dodatečně je plně podporováno kolem 30 XEP standardů například XEP-0054 vcard-temp a další.

Kapitola 3

Data mining

cite z co je v thek kappitole

Anglický filozof Bacon kdysi řekl, že ve znalostech je síla (Knowledge itself is a power.

Na úvod kapitoly jsou definovány základní pojmy, jez zde budou používány. Mezi základní patří data mining neboli česky.... Definic tohoto výrazu se v odborné literatuře oběvuje několik. Zde uvedená je kombinací dvou „definic“[...].

Historický 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 například odvětví zdravotnictví, vědě, marketingu (efektivita reklamních kampaní, segmentace zákazníků) a další. 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 kategorií :

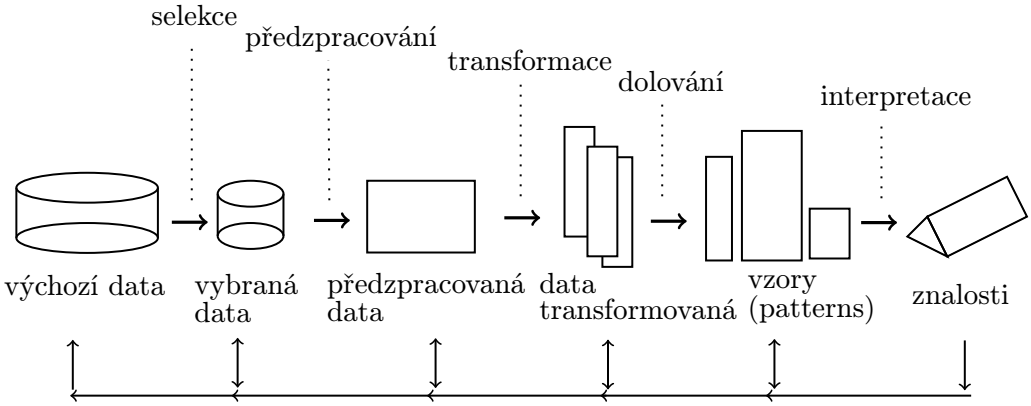
1. počet záznamů nebo–li objektů v databázi
2. počet polí nebo–li atributů objektů v databázi

Proces data mining je pouze jedna část odvětví nazývané dobývání znalostí z databází nebo–li KDD (dále budeme používat již zkratku KDD) [3] definované níže 3.0.1. Vznik disciplíny je důsledkem nepřeberného množství automaticky sbíraných dat a potřeba tato data dále využívat. Podstatným znakem je správnost reprezentace výsledků formou uživateli nejbližší jako je implikace ve tvaru rozhodovacích pravidel, asociačních pravidel, rozhodovací stromy, shluky podobných dat a další. Základem KDD je praktická použitelnost metod, očekává se zjištění nových skutečností namísto prezentování známých informací.

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

[1] International Joint Conference on Artificial Intelligence
[2] Association for the Advancement of Artificial Intelligence
[3] Knowledge Discovery in Database

Grafické znázornění definice 3.0.1 je popsáno schématem na obrázku 3.2, který ukazuje časový harmonogram v KDD. Schéma znázorňuje následnost jednotlivých procesů, které tvoří KDD. KDD je iterativní proces, skutečnosti nalezené v předešlých části zjednodušují a zpřesňují vstupy pro následující fáze. Jakmile jsou získány znalosti, jsou předvedeny uživateli. Pro přesnost může být část procesu KDD ještě upřesněna, a tím získány „přesnější a vhodnější“ výsledky.



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

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

- **čištění dat** – fáze, ve které jsou nepodstatné údaje odstraněny z kolekce.
- **integrace dat** – data z více zdrojů,často heterogenní, jsou kombinována 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 3.2.

Základní pojmy

V této podsekci jsou ve stručnosti vysvětleny základní nejdůležitější pojmy, kterých se bude v práci dále využívat.

Definic 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 [13].

Mezi další četně 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 [7].

Informace definované v 3.0.4, jsou zpracovaná data interpretována uživatelem.

Definice 3.0.4 Informace jsou data, která mají sémantiku (význam) [7].

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 dovozené informace.

Definice 3.0.5 Znalosti jsou informace po jejich zařazení do souvislostí [7].

3.1 Metody dolovaní 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 s 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 některých z nich nejčastějších:

Asociační pravidla

Asociační pravidla jsou založena na syntaxi IF-THEN. Jejich rozšíření se datuje do 90.let 20.století. Pan Agrawal jej 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 príkladů 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 terminilogie:

```
Ant => Suc
```

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

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

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

	Suc	**¬Suc**	$\sum$
Ant	a	b	r
¬Ant	c	d	s
$\sum$	k	l	n

Tabulka 3.1: Kontingenční tabulka [2].

Mezi základní charakteristiky asociačních pravidel podle Agrewalova 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 reprezentující 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. Instance jsou tvořeny vzorky dat, 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

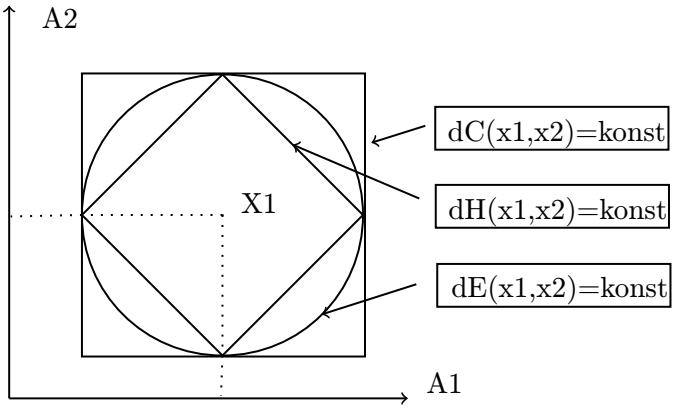
Základem modelů jsou trénovací data. Příklad některých klasifikačních modelů :

- 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, ze data v minulosti budou chovat stejně nebo aspoň 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



Obrázek 3.2: Proces dobývání znalostí z databází podle knihy autora Fayyad [3].

e test data set se determine how many purchases klasifikaci. Clustering (Segmentation) - na clustering to lze dívat jako na automatickou included both items in each two-item

set. Again, a do tzv. clusteru of support is který se co Algoritmy této metody seskupují podobné objekty minimum level (shluku dat), required. In Figure 13-15, you can see we have 5 two-item sets with the minimum nejvíce required. ? r r support liší od clusteru ostatních. Tyto clustery nejsou pˇredem definované a je tak na pˇrísluš- Items from the two-item sets are now combined urˇform three-item sets. This r ˇ ném analytikovi, aby je zkoumal a pokoušel se najít to cité závislosti. Pokud jsou pˇredmetem process continues until there is either one or zero sets with the minimum support. In ˇ zkoumání zákazníci, mluvíme vetšinou o tzv. segmentaci. Figure 13-15, no three-item sets have the minimum support required so, in this case, ˇ Proces clusteringu je také schopen odhalit nejˇ astejší c ˇ the algorithm does not continue with four-item sets. posloupnosti mezi daty. Proto je casto Once the sets are created, the algorithm creates ? membership rules based nebo využíván napˇríklad k mapování chování zákazníku na webových stránkáchon the také k urr result. The algorithm determined that 16,044 purchases included the British Tank ˇ ˇ cení sledovanosti televizních poˇ ? Commander. Of those purchases, radu v závislosti na jejich vysílacím case. Clustering je 15,232, or 94.9také používán k odhalení for predicting future ci závislostí In the future, when ? ˇ ? Driver. This becomes a rulejakýchkoliv problému associations. a muže tak sloužit jako vstup someone puts the British miningu. Pˇríslušné algoritmy k této metode jsou nazývány100, Tank Commander in their shopping cart, ˇ times out of Cluste- 95 pro ostatní metody data r they will also include the Germa

je pak další typ, který je podobný klasifikaci. V tomto případě ale nejsou předem určené skupiny, do kterých se data zařazují, ale cílem je odhalení skupin nebo shluků dat s podobnými atributy a interpretace těchto míst hustšího výskytu instancí. Problémy tohoto typu se velice často řeší v oblasti marketingu, například při určování skladby trhu, kdy se identifikují cílové skupiny zákazníků a vlastnosti, které danou skupinu charakterizují.

V případě nejbližšího souseda jsou koncepty (třídy) reprezentovány svými typickými představiteli. V procesu klasifikace se pak nový příklad zařadí do třídy na základě podobnosti (nejmenší vzdálenosti k reprezentantovi nějaké třídy ? viz obr. 8). Jde tedy o metodu která vychází ze shlukové analýzy. Klíčovým pojmem je koncept podobnosti, resp. vzdálenosti dvou příkladů.

rozdělení databází, takže záznamy jsou seskupovány podle obdobných charakteristik (komodit, lokalit, objemů prodeje aj.)

3.2 shlukování

Segmentace: Někdy se zaměřujeme na dělení objektů do předem neurčených skupin. Zajímá nás, zda existují skupiny zákazníků, jež jsou nějakým způsobem specifické a odlišné od ostatních skupin, a snažíme se tyto skupiny určit.

3.3 Dolovaní znalosti z databází

3.4 Programy

Kapitola 4

Developer

4.1 Databáze

PostgreSQL

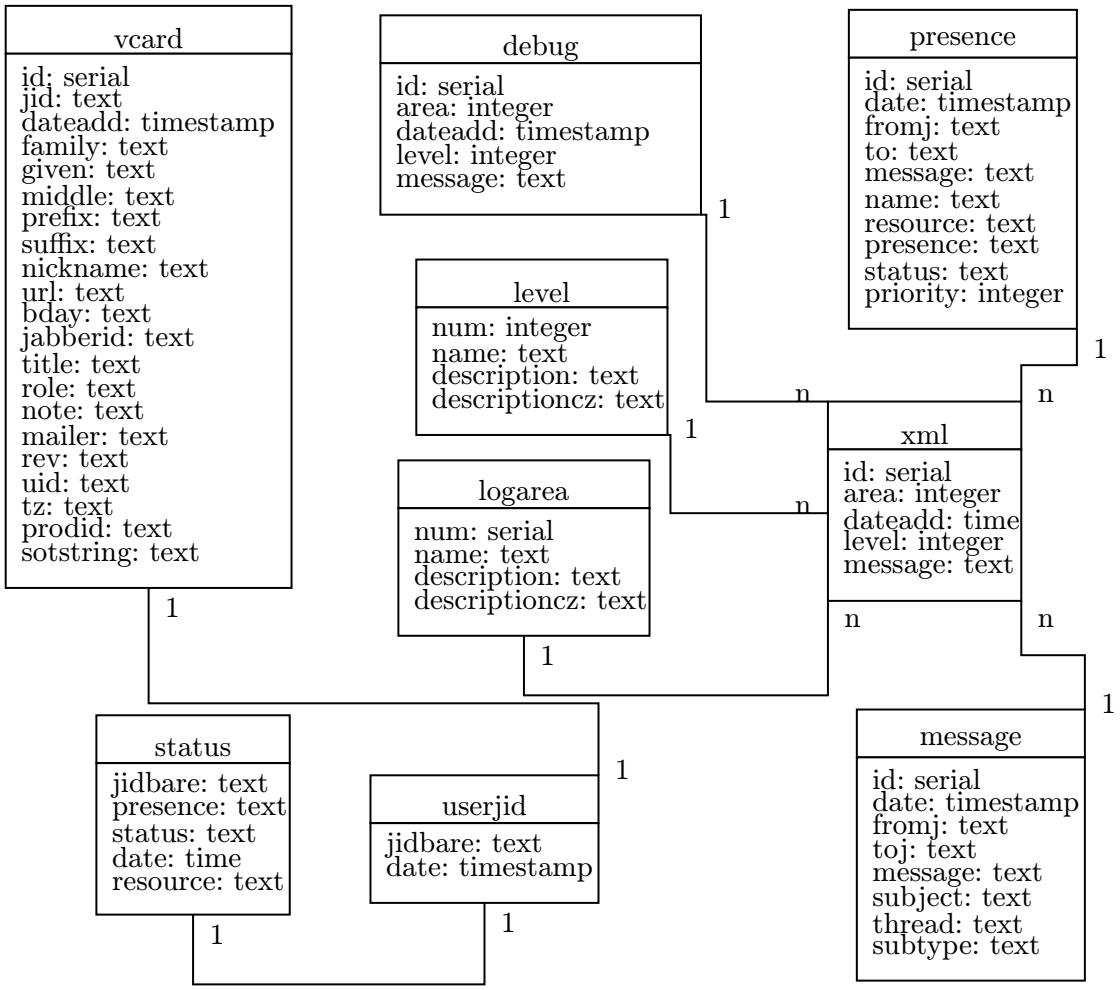
Návrh databáze

4.2 Bot

Návrh bota

4.3 Knihovny, jazyk

4.4 Jiné produkty



Obrázek 4.1: Struktura databáze

Kapitola 5

Závěr

Literatura

[1] Adams, D.: *Programming jabber*. Sebastopol: O'Reilly, první vydání, 2002, 455 s., iSBN 05-960-0202-5.

[2] Berka, P.: *Dobývání znalostí z databází*. Praha: Academia, první vydání, 2003, 366 s., iSBN 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., iSBN 02-625-6097-6.

[4] Fred, H.: *Computer networking and the internet*. Edinburg: Addison-Wesley Publishing Company, první vydání, 2005, 803 s., iSBN 03-212-6358-8.

[5] Hildebrand, J.; Saint-Andre, P.: XEP-0080: User Location. [online], 15-09-2009, [cit. 11. dubna 2011].
URL http://xmpp.org/extensions/xep-0080.html

[6] Hildebrand, J.; Saint-Andre, P.; Tronçon, R.; aj.: XEP-0115: Entity Capabilities. [online], 02-26-2008, [cit. 11. dubna 2011].
URL http://xmpp.org/extensions/xep-0115.html

[7] Hruška, T.: *Informační systémy : IIS/PIS*. Brno: Fakulta informačních technologií, 2008, 14733 s.

[8] Kolektiv autorů: Extensible Markup Language (XML) 1.0. [online], 26-11-2008, [cit. 11. dubna 2011].
URL http://www.w3.org/TR/2008/REC-xml-20081126/

[9] Kosek, J.: *XML pro každého : podrobný průvodce*. Praha: Grada, první vydání, 2000, 163 s., iSBN 80-716-9860-1.

[10] Kurose, J. F.; Ross, K. W.: *Computer networking : top-down approach featuring the internet*. Boston: Addison-Wesley Publishing Company, druhé vydání, 2003, 752 s., iSBN 03-211-7644-8.

[11] Millard, P.; Saint-Andre, P.; Meijer, R.: XEP-0060: Publish-Subscribe. [online], 12-07-2010, [cit. 11. dubna 2011].
URL http://xmpp.org/extensions/xep-0060.html

[12] Moore, D.; Wright, W.: *Jabber developer's handbook*. Indianapolis: Sams Publishing, první vydání, 2004, 487 s., iSBN 06-723-2536-5.

[13] Nemrava, M.; Pospíšil, J.: Dolování dat a jeho aplikace. [online], 2006, [cit. 11. dubna 2011].
URL http://www.spatial.cs.umn.edu/paper_ps/dmchap.pdf

[14] Saint-Andre, P.: XEP-0118: User Tune. [online], 30-01-2008, [cit. 11. dubna 2011].
URL http://xmpp.org/extensions/xep-0118.html

[15] Saint-Andre, P.; Smith, K.: XEP-0163: Personal Eventing Protocol. [online], 12-07-2010, [cit. 11. dubna 2011].
URL http://xmpp.org/extensions/xep-0163.html

[16] Saint-André, P.: Extensible Messaging and Presence Protocol (XMPP): Core. [online], 10-2004, [cit. 11. dubna 2011].
URL http://tools.ietf.org/html/rfc3920

[17] Saint-André, P.: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence. [online], 10-2004, [cit. 11. dubna 2011].
URL http://tools.ietf.org/html/rfc3921

[18] 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., iSBN 978-059-6521-264.

[19] Stevens, W.; Fenner, B.; M.Rudoff, A.: *UNIX Network Programming*. Boston: Addison-Wesley Publishing Company, třetí vydání, 2004, 991 s., iSBN 01-314-1155-1.

Příloha A

Obsah CD

Příloha B

Manual

Příloha C

Konfigrační 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
</ig>
```

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]/>
  <subject>  </subject>
  <thread>  </thread>
  <error>   </error>
  <x>   </x>
</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/*>.