

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

## 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 POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

## **DATAMINING Z JABBERU**

DATAMINING FROM JABBER

### **BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

### **AUTOR PRÁCE**

AUTHOR

JAROSLAV SENDLER

### **VEDOUCÍ PRÁCE**

SUPERVISOR

Ing. JOZEF MLÍCH

BRNO 2011

SEM VLOZTE  
ZADANI

PRACE

## Abstrakt

Tato bakalářská práce se zabývá získáváním a analýzou dat nabytých z protokolu Jabber. Je zde popsán návrh klienta, který sbírá data a následně je ukládá do databáze. Tento klient plní funkci robota, který vhodně reaguje na zvolené dotazy. Před analýzou nashromážděných informací je provedena jejich transformace, která zohledňuje časové vlastnosti dat. Dále jsou popsány postupy při dolování z dat a algoritmus  $k$ -means s nástrojem RapidMiner. Výsledky získané procesem data mining jsou dále vhodně upraveny a prezentovány v grafické formě i pomocí tabulky.

## Abstract

This thesis deals with the collection and analysis of data acquired from the Jabber protocol. There is a proposal described a client who collects data and then stores them in a database. The client performs the function of a robot that responds appropriately to the selected questions. Before analyzing the information gathered is made of a transformation that takes into account the temporal characteristics of data. The following describes the procedures for data mining and  $k$ -means algorithm with RapidMiner tool. The results obtained by the process of data mining are also to be adjusted and presented in graphical form and using a spreadsheet.

## Klíčová slova

Jabber, XMPP, robot, data mining, shlukování,  $k$ -means, dolování z dat, transformace dat, RapidMiner.

## Keywords

Jabber, XMPP, robot, data mining, clustering,  $k$ -means, data transformation, RapidMiner.

## 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. Jozefa Mlícha. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Jaroslav Sendler

16. května 2011

## Poděkování

Tímto bych chtěl poděkovat mému vedoucímu bakalářské práce Ing. Jozefovi 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

|                                      |           |
|--------------------------------------|-----------|
| <b>1 Úvod</b>                        | <b>5</b>  |
| <b>2 XMPP</b>                        | <b>7</b>  |
| 2.1 Architektura . . . . .           | 7         |
| 2.2 XML . . . . .                    | 10        |
| 2.3 Stanza . . . . .                 | 11        |
| 2.4 Rozšíření . . . . .              | 13        |
| <b>3 Data mining</b>                 | <b>16</b> |
| 3.1 Transformace dat . . . . .       | 18        |
| 3.2 Shlukování . . . . .             | 21        |
| 3.3 Programy . . . . .               | 25        |
| <b>4 Implementace</b>                | <b>27</b> |
| 4.1 Architektura . . . . .           | 27        |
| 4.2 Návrh databáze . . . . .         | 28        |
| 4.3 Návrh robota . . . . .           | 30        |
| 4.4 Pokračování práce . . . . .      | 31        |
| <b>5 Vyhodnocení výsledků</b>        | <b>33</b> |
| 5.1 Manuální rozbor dat . . . . .    | 33        |
| 5.2 Programové vyhodnocení . . . . . | 34        |
| <b>6 Závěr</b>                       | <b>38</b> |
| <b>A Obsah CD</b>                    | <b>42</b> |
| <b>B Slovník zkratk</b>              | <b>43</b> |
| <b>C Stanza - základní schéma</b>    | <b>44</b> |
| <b>D Návrh databáze</b>              | <b>48</b> |
| <b>E Přehled příkazů robota</b>      | <b>49</b> |
| <b>F Přehled vztahů uživatelů</b>    | <b>50</b> |

# Seznam obrázků

|     |                                                                            |    |
|-----|----------------------------------------------------------------------------|----|
| 2.1 | Distribuovaná architektura Jabber. . . . .                                 | 8  |
| 2.2 | Rozebraná struktura Jabber ID. . . . .                                     | 9  |
| 3.1 | Proces dobývání znalostí z databází podle knihy autora Fayyad [4]. . . . . | 17 |
| 3.2 | Srovnání výpočtu vzdáleností od bodu $x_1$ [2]. . . . .                    | 22 |
| 3.3 | Algoritmus $k$ -means zobrazený pomocí konečného automatu. . . . .         | 24 |
| 4.1 | Struktura architektury bakalářské práce. . . . .                           | 28 |
| 4.2 | Vybraná část struktury databáze. . . . .                                   | 29 |
| 4.3 | Vybraná část struktury robota. . . . .                                     | 31 |
| 5.1 | Struktura procesu v nástroji RapidMiner. . . . .                           | 35 |
| 5.2 | Výsledné shluky převedeny do 3D prostoru. . . . .                          | 35 |
| 5.3 | Průběh statusů shluků podle času. . . . .                                  | 36 |
| C.1 | Ukázka „šíření“ <i>User Tune</i> . . . . .                                 | 45 |
| D.1 | Struktura databáze . . . . .                                               | 48 |

# Seznam tabulek

|     |                                                              |    |
|-----|--------------------------------------------------------------|----|
| 2.1 | Přehled Jabber serverů. . . . .                              | 10 |
| 3.1 | Ukázka tabulky <i>presence</i> . . . . .                     | 20 |
| 3.2 | Ukázka modifikované tabulky <i>presence_modify</i> . . . . . | 20 |
| 5.1 | Přehled manuálního rozboru dat. . . . .                      | 34 |
| 5.2 | Procentuální shoda jednotlivých uživatelů. . . . .           | 37 |
| E.1 | Přehled příkazů pro robota. . . . .                          | 49 |
| F.1 | Zobrazení uživatelského JID na čísla. . . . .                | 50 |
| F.2 | Procentuální shoda jednotlivých uživatelů. . . . .           | 51 |



# Příklady

|     |                                                                                                                    |    |
|-----|--------------------------------------------------------------------------------------------------------------------|----|
| 2.1 | Ukázka základního XML dokumentu. . . . .                                                                           | 11 |
| 2.2 | Použití elementu <i>message</i> . . . . .                                                                          | 11 |
| 2.3 | Použití elementu <i>iq</i> . . . . .                                                                               | 12 |
| 2.4 | Použití elementu <i>presence</i> . . . . .                                                                         | 12 |
| 2.5 | Začátku vysílání rozšířeného statusu. . . . .                                                                      | 13 |
| 2.6 | Dotaz na podporované protokoly. . . . .                                                                            | 14 |
| 3.1 | Metoda <i>k</i> -means byla převzata z [2]. . . . .                                                                | 25 |
| C.1 | Popis elementu <i>iq</i> . . . . .                                                                                 | 44 |
| C.2 | Popis elementu <i>message</i> . . . . .                                                                            | 44 |
| C.3 | Popis elementu <i>presence</i> . . . . .                                                                           | 45 |
| C.4 | Informování serveru o právě přehrávající hudbě. . . . .                                                            | 46 |
| C.5 | Server informuje uživatele podporující rozšíření o stavu <i>user@jabber.com</i> . . . . .                          | 46 |
| C.6 | Server přepośle informace o přehrávané hudbě všem otevřeným spojením<br>uživatele <i>user@jabber.com</i> . . . . . | 47 |
| C.7 | Uživatel ukončil „vysílání“ rozšířených zpráv o svém stavu. . . . .                                                | 47 |
| C.8 | Server informuje klienty o ukončení šíření rozšířeného statusu. . . . .                                            | 47 |

# 1 Kapitola 1

## 2 Úvod

3 V současné době jsou řazeny k nejrozšířenějším psaným dorozumívacím prostředkům real-  
4 time komunikační sítě. Samozřejmě je tento jev k vidění až koncem minulého desetiletí.  
5 Psaná komunikace je již několik století využívána jako prostředek k dorozumívání lidí mezi  
6 sebou. Za toto období prošla výrazným pokrokovým vývojem, proto lze u ní nalézt mnoho  
7 časových mezníků, které ji výrazně ovlivnily. Jako příklad jednoho z prvních komunikačních  
8 kanálů lze uvést posly, kteří často nesly zprávy na vzdálenosti několika kilometrů. Dalším  
9 výrazným prvkem ve vývoji dorozumívacích prostředků bylo zavedení pošty a objevení  
10 telegrafu.

11 Příchodem internetu nastal v komunikaci zásadní zlom. Postupným rozšířením pokrytí  
12 a dostupnosti této technologie začalo vznikat mnoho nových komunikačních prostředků.  
13 Příkladem mohou být elektronické zprávy, RSS zprávy nebo real-time komunikační sítě.  
14 Právě poslední zmíněná metoda zažívala v moderní době velký růst v oblasti popularity,  
15 ať už v podobě ICQ protokolu nebo dnes velmi rozšířené sociální sítě Facebook. Jedna  
16 z hlavních výhod těchto služeb, například v porovnání s klasickou poštou, je jejich rychlost  
17 doručení. Naproti běžné elektronické poště jsou uživatelé informováni o stavu příjemce  
18 zprávy. Real-time komunikační prostředky nabízejí služby, kterými je možné zjistit, zda  
19 se příjemce zprávy nachází u dorozumívacího zařízení. Díky této schopnosti je uživateli  
20 umožněno zasílat zprávy a obratem na ně očekávat odpovědi.

21 S přibývajícími elektronickými daty, na poli ať už vědních nebo praktických oborů,  
22 přichází potřeba tyto informace uchovávat. K těmto účelům slouží databáze, které se svou  
23 strukturou zaměřují především na snadnou kontrolu dat, vyhledávání a jejich analyzování.  
24 S rostoucím obsahem se ale uložené informace stávají pouze daty bez významu. Proto  
25 vnikla nová disciplína nazvaná *data mining*, která si klade za cíl znovunalezení „ztracených“  
26 informací (na první pohled neviditelných) nebo nalezení informací úplně nových.

27 Předmětem této bakalářské práce je seznámení se s problematikou komunikace probíha-  
28 jící přes Jabber síť. Konkrétním cílem je vytvoření jednoduchého Jabber klienta, který by  
29 byl schopen získávat statistická data. Nashromážděná data jsou dále využita pro pozdější  
30 analýzu a grafickou reprezentaci informací z nich získaných. Hlavním cílem této práce je  
31 získat neznámé informace z real-time komunikační sítě Jabber.

32 Technická zpráva je tvořena z šesti kapitol. Kapitola **druhá** je tvořena popisem proto-  
33 kolu XMPP, na kterém je postavena real-time komunikační služba Jabber. Je zde popsána  
34 architektura sítě a základní stavební kameny XMPP protokolu, které jsou využívány sítí  
35 Jabber jako nástroj k zprostředkování jednotlivých služeb. V závěru tohoto oddílu je vě-  
36 nována část vybraným standardům, které rozšiřují základní XMPP protokoly. Jsou zde  
37 uvedena pouze ta rozšíření, která byla po konzultaci s vedoucím práce, označena za rele-

38 vantní k této práci.

39     Obsah **třetí** kapitoly je zaměřen na popis procesu data mining. Zabývá se začleněním  
40 této metody do komplexnějšího procesu, který je nazýván získávání znalostí z databází. Da-  
41 lší součást této kapitoly se zabývá nezbytnými kroky transformace dat, která jsou získána  
42 z Jabber komunikace. Při tomto procesu jsou data převedena do vhodné podoby pro proces  
43 dolování z dat. Následuje část, kterou jsou popsány vybrané metody dolování dat a také je  
44 zde podrobně rozebrán algoritmus  $k$ -means. Tento algoritmus je využit pro samotné dolo-  
45 vání z dat, které je zprostředkováno aplikací RapidMiner. Popis tohoto programu a dalších  
46 vybraných nástrojů uzavírá třetí kapitolu.

47     Implementační část této práce je popsána kapitolou **čtvrtou**. Je členěna na oddíly,  
48 které odpovídají vybraným částem architektury této práce. Elementy architektury, které  
49 jsou popsány vlastní podkapitolou, jsou rozšířeny o zjednodušený návrh struktury v podobě  
50 grafických schémat. Závěr této kapitoly se zamýšlí nad rozšířeními robota a nad dalším  
51 pokračováním této práce.

52     Výsledky získané touto prací jsou prezentovány v kapitole **páté**. Je zde ukázán model  
53 používaný programem RapidMiner pro dobývání znalostí. Konkrétně jde o shlukování uži-  
54 vatelů sítě Jabber do skupin podle času stráveném v samotné síti a jejich stavu. Výsledky  
55 jsou představeny pomocí různých grafických entit, ať už v podobě diagramu nebo grafu,  
56 který je transformovaný do dvourozměrného prostoru.

57     Nemalý informativní obsah tvoří i **přílohy**. V první řadě je to slovník zkratk, shrnující  
58 slovní spojení z celého tohoto dokumentu. Následuje podrobnější popis základních entit  
59 stanzy a přehled průběhu rozšíření. V další části je ukázán kompletní návrh databáze,  
60 který doplňuje základní popis ze čtvrté kapitoly. Přehled příkazů, na které robot vytvořený  
61 v této práci reaguje, je ukázán v poslední části příloh.

## 62 Kapitola 2

# 63 XMPP

64 V následující kapitole jsou, pro usnadnění a jednodušší pochopení, rozebrány základní sta-  
65 vební kameny protokolu Extensible Messaging and Presence Protocol (XMPP). Konkrétně  
66 jsou zde popsány stávající vlastnosti implementace, architektura protokolu XMPP obecně  
67 [23, 24] a další detaily protokolu [1, 25, 14]. Vzhledem k požadavkům na dolování v da-  
68 tech popsaných v následující kapitole je kladen důraz na vybraná rozšíření [22, 12]. Tato  
69 rozšíření tvoří základ pro některé rozšířené statusy, jako je například User Tune [18], User  
70 Mood [21], User Location [6] a další. Další informace použité pro popis a pochopení XML  
71 jazyka byly čerpány z [10, 9].

72 Vznik samotného protokolu XMPP je datován do roku 2004 (březen), kdy na něj byl  
73 přejmenován Jabber. Původní projekt Jabber byl vytvořen roku 1998 autorem Jeremie  
74 Millerem, který ho založil za účelem vytvořit svobodnou otevřenou IM službu. Uvedený  
75 projekt měl obsahovat tři základní vlastnosti, do kterých se zahrnují jednoduchost a sro-  
76 zumitelnost pro implementaci, jednoduchost v oblasti šíření a otevřenost podobě veřejně  
77 dostupného popisu samotného protokolu. Základní vlastnosti a výhody klientů a serverů  
78 budou podrobněji popsány níže. Roku 1999, 4. ledna byl vytvořen první server se jmé-  
79 nem Jabber. Komunita vývojářů se chopila iniciativy a vytvořila klienty, kteří dokázali se  
80 serverem komunikovat, pro různé platformy (Linux, Macintosh, Windows). Roku 2004 byl  
81 protokol XMPP přidán mezi RFC<sup>1</sup> dokumenty. Základní norma popisující obecnou struk-  
82 turu protokolu je RFC 3920 [23] a RFC 3921 [24], který se zaměřuje na samotný instant  
83 messaging a zobrazení stavu. Další zdokumentovaná rozšíření jsou vydávána v podobě tzv.  
84 XEP (XMPP Extension Protocol) dokumentů, které jsou známé také pod starším názvem  
85 JEP (Jabber Enhancement Proposal). Dnešní počet těchto norem se blíží k číslu 300. Každý  
86 XEP obsahuje stav vývoje (schválení), ve kterém se zrovna nachází.

87 Jako bezpečnostní prvky jsou zde podporovány SASL, TLS a GPG. XMPP protokol  
88 je postaven na obecném značkovacím jazyce XML, proto vlastnosti popsané dále v této  
89 kapitole platí i pro tento protokol.

## 90 2.1 Architektura

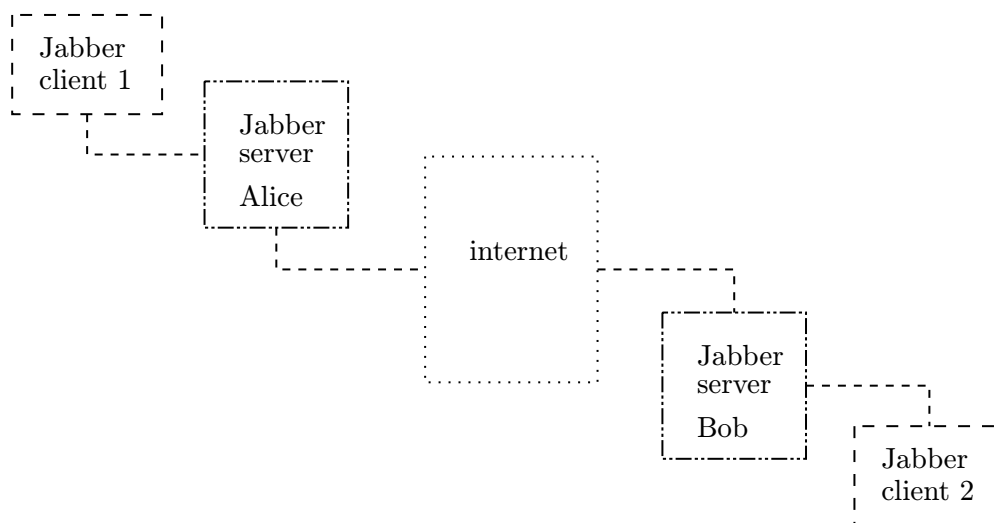
91 Dobře navržená internetová technologie je tvořena správně fungujícími komponenty, které  
92 mezi sebou dokáží vytvořit spojení a následně započít komunikaci. Pro popis Jabber ar-  
93 chitektury v této práci bylo čerpáno z [1, 25]. Tato struktura se nejvíce podobá struktuře  
94 posílání e-mailů. Hlavní předností Jabber sítě je, tak jako u elektronické pošty, její decentra-

---

<sup>1</sup>RFC request of comments — žádost o komentáře

lize. V případě Jabberu je decentralizace chápána jako možnost provozovat vlastní server, na rozdíl od jiných komunikačních systémů jako je například Facebook, kde existuje pouze jediný poskytovatel služby. V případě serveru je kladen důraz na spolehlivost a rozšiřitelnost a u klienta na uživatele. Každý server pracuje samostatně, což znamená, že chod ani výpadek jiné datové stanice žádným způsobem jeho běh neovlivní. V případě výpadku jiného serveru bude nedostupný pouze seznam kontaktů a služeb, které registrovaným uživatelům poskytoval.

Obrázek 2.1 znázorňující distribuovanou architekturu Jabberu byl převzat z [1] a doplněn o názvy jednotlivých komponent. Komunikace dvou Jabber klientů probíhá za účasti jejich serverů a sítě, která je spojuje. Spojení mezi nimi bývá často šifrováno.



Obrázek 2.1: Distribuovaná architektura Jabber.

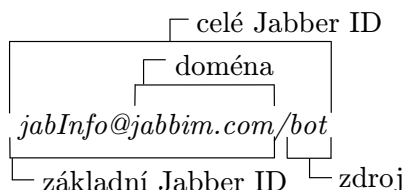
Architektura Jabber serverů využívá velké množství mezi-doménových připojení podobně jako internetový systém elektronické pošty. Komunikace klienta z jedné domény s klientem z jiné na rozdíl od e-mailového modelu nevyžaduje spolupráci třetích stran. Klient se spojí s „domácím“ serverem, který přímo naváže spojení se serverem požadovaného klienta. Tyto vlastnosti jsou zárukou pro bezpečný přenos zpráv, znemožňující „krádeže“ JID<sup>2</sup>, který je popsán níže, a spamování.

## Jabber ID

Jabber ID (JID) je jednoznačný virtuální identifikátor uživatele na síti. V případě založení účtu nejsou rozlišována velká a malá písmena, což znamená, že Jabber není case-sensitive. Jednoznačný Jabber identifikátor je složen ze dvou částí: *Jabber bare* neboli čisté ID a *resource* [23]. Základní část na první pohled připomíná e-mailovou adresu *user@server*. Druhá část slouží k přesné identifikaci jednotlivých spojení. Je použita ke směrování síťového provozu s uživateli v případě otevření většího množství spojení pod jedním uživatelem. Společně Jabber bare a resource tvoří tzv. *full JID* — *user@server/resource* například *jabInfo@jabim.cz/bot*. Jednotlivé části uživatelského jména popsané v tomto odstavci jsou ukázány v obrázku 2.2.

<sup>2</sup>uživatelské jméno

121 Další vymoženost JabberID oproti e-mailové adrese je jeho možnost používat prakticky  
libovolné národní znaky u doménových jmen a uživatelských účtů [25]. Využíváním kó-



Obrázek 2.2: Rozebraná struktura Jabber ID.

122  
123 dování UNICODE, se XMPP stává plně mezinárodní a není jako jiné protokoly omezen  
124 rozsahem ASCII tabulky. Přestože je tato vymoženost k dispozici, doposud není žádným  
125 výrazným způsobem využívána.

## 126 Klient

127 Klient je často jednoduchá aplikace pracující se vzdálenými službami, které jsou provo-  
128 vány serverem. V této práci je zastoupen robotem s konzolovým rozhraním. XMPP svou  
129 architekturou nutí, aby byl co nejjednodušší. Vlastnosti, které by měl mít, jsou shrnuty  
130 podle [14] do tří bodů:

- 131 1. komunikace s jedním Jabber serverem pomocí TCP socketu, který garantuje spolehlivé  
132 doručení zpráv na rozdíl od UDP. Nad tímto transportním protokolem dále běží  
133 kryptografický protokol TLS, který zabezpečuje komunikaci klient-server a server-  
134 server.
- 135 2. rozparsování a následná interpretace příchozí XML zprávy „stanza“ (kapitola 2.3).
- 136 3. porozumění sadě zpráv (*message*, *iq*, *presence*) z Jabber jádra [23].

## 137 Server

138 Informace použité pro popis XMPP serveru byly čerpány z [14]. K hlavním charakteris-  
139 tikám serveru oproti klientovi, jehož základní vlastností byla jednoduchost, patří stabilita  
140 a bezpečnost. Je pro něj vyhrazen TCP port 5222. Komunikace mezi servery je realizována  
141 přes port 5269. Každý server uchovává seznam zaregistrovaných uživatelů, který nevykazuje  
142 žádný jiný server. Zaregistrovaní uživatelé v daném seznamu se mohou do sítě připojovat  
143 pouze přes něj. To zajišťuje nemožnost „krádeže“ účtu. Protože XMPP komunikace probíhá  
144 přes síť, musí mít každá entita adresu, v tomto případě nazvána JabberID. XMPP spoléhá  
145 na DNS což znamená, že používá jména na rozdíl od IP protokolu.

146 Server Jabber je systém spravující tok dat mezi jednotlivými komponentami, které spo-  
147 lečně tvoří Jabber služby. Například *Jabber Session Manager* (JSM) poskytne funkce pro  
148 IM komunikaci a práci se seznamem kontaktů. Komunikace mezi jednotlivými servery, jak  
149 je uvedeno na obrázku 2.1, je zprostředkována za pomoci komponenty *S2S* (server to ser-  
150 ver). Při připojení klienta k serveru je komunikace řízená pomocí *C2S* (client to server).  
151 Jak již bylo řečeno, Jabber síť využívá doménová jména místo špatně zapamatovatelných  
152 IP adres. Pro tento způsob identifikace je určena služba *dnsrv*, která se stará o překlad  
153 názvů. V podstatě je to komponenta, která zajišťuje směrování paketů na jiný server.

V tabulce 2.1 jsou shrnuty informace o serverech Jabberu. První sloupec tvoří jméno, následuje programovací jazyk, v němž je napsán. Většina aplikací pro servery je vydávána pod licenci GPL<sup>3</sup>. U všech aplikací byla zkoumána nejaktuálnější verze. Její číslo lze nalézt 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 programů pro server Jabber jsou stále vyvíjeny, tedy kromě jabberd14. Hlavním účelem tabulky je prezentovat důležité vlastnosti serverů v oblasti podpory rozšířených statusů. Jedná se o standardy *pubsub*<sup>4</sup> (XEP-0060) [12] a o jeho verzi, která je více zaměřena na uživatele *pep*<sup>5</sup> (XEP-0163) [22]. Obě tato rozšíření tvoří nezbytnou základnu pro *rozšířené statusy* a proto je jejich podpora jak u serverů, tak klientů vyžadována. Podrobněji toto téma bude rozebráno v některé následující podkapitole.

| Server    | Jazyk       | Verze   | XEP-0060 | XEP-0163 |
|-----------|-------------|---------|----------|----------|
| ejabberd  | Erlang/ Top | 2.1.6   | ANO      | ANO      |
| Openfire  | java        | 3.6.4   | ANO      | ANO      |
| jabberd2  | c           | 2.2.11  | NE       | NE       |
| jabberd14 | c, c++      | 1.6.1.1 | ANO      | NE       |
| Prosody   | lua         | 0.7.0   | NE       | ANO      |
| Tigase    | java        | 5.0.0   | ANO      | ANO      |

Tabulka 2.1: Přehled Jabber serverů.

Z výše uvedené tabulky je zřejmé, že aplikace pro servery, které jsou stále ve vývoji, podporují tzv. *rozšířené statusy*. Tedy kromě programu jabberd2.

## 2.2 XML

Jazyk XML (eXtensible Markup Language) [9], 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, které můžeme navzájem zanořovat. Vyznačujeme je pomocí značek — počáteční a ukončovací. Pomocí tohoto jazyka je tvořena *stanza* popsána v následující kapitole.

Ukázka možné struktury dokumentu psaného jazykem XML je zobrazena na příkladu 2.1. Standardně je předpokládáno, že je psán v kódování UTF-8 [10], ale je-li jako v tomto případě použito jiné, musí být konkrétní kódování uvedeno na jeho počátku. V opačném případě nemusí být obsah správně zobrazen. Na začátku dokumentu se také uvádí verze XML, ve které je dokument psán (1. řádek příkladu). Následuje kořenový element, který je uzavřen na samotném konci dokumentu. 4. řádek prezentuje možnost použití prázdného elementu, který obsahuje jeden atribut s názvem zkratky fakulty. Velký význam zde mají úhlové závorky. Jsou jimi z obou stran obaleny všechny elementy.

<sup>3</sup>General Public License — všeobecná veřejná licence GNU

<sup>4</sup>Publish-Subscribe

<sup>5</sup>Personal Eventing Protocol

---

```

1      <?xml version="1.0" encoding="iso-8859-2"?>
2      <fakulta>
3          <název>Fakulta informačních technologií</název>
4          <zkratka fakulty="FIT"/>
5          <typy studia>
6              <bakalářské titul="Bc."></bakalářské>
7              <magisterské></magisterské>
8              <doktorské></doktorské>
9          </typy studia>
10     </fakulta>

```

---

Příklad 2.1: Ukázka základního XML dokumentu.

## 2.3 Stanza

Základní jednotkou pro komunikaci založenou na XML je stanza. Z jednoduššího pohledu je možné se na ni dívat jako na jeden dlouhý XML soubor. Při zahájení komunikace se tento soubor „otevře“. Jeho samotné uzavření probíhá až při odhlášení od sítě, neboli přepnutí klienta do stavu offline. Stanzu je tedy možné vnímat jako stream, který obsahuje všechna data probíhající komunikace. Mezi elementy používané pro komunikaci klienta se serverem patří tyto tři: *message*, *presence* a *iq*. Každý zde uvedený člen má svůj jednoznačný význam. V následujících odstavcích jsou jednotlivé části stanzy blíže definovány a na reprezentativních příkladech jsou ukázány jejich základní struktury a možnosti využití v praxi.

První prvek, který bude charakterizován je označen anglickým výrazem *message* (zpráva). Jak již název napovídá, slouží k posílání zpráv všeho druhu. Je to základní metoda pro rychlý přenos informací z místa na místo. Zprávy jsou typu „push“, což znamená, že jsou odeslány a není očekávána žádná aktivita od příjemce, která by přijetí potvrdila. Jedno z dosavadních využití se nachází v klasické komunikace po internetu, tzv. instant messaging (IM). K dalším možným použitím patří skupinový chat a oznamovací nebo upozorňující zprávy. Každá z těchto zpráv je tvořena z minimální povinné struktury. Tak jako u klasické poštovní korespondence nesmí chybět adresa odesílatele a adresa příjemce, kterému je zpráva adresována. Podle možnosti použití jsou zprávy děleny do kategorií. Jmenovitě toto rozdělení implementuje atribut *type*, který může nabývat jednu ze čtyř hodnot. Jsou rozlišovány zprávy pro komunikaci mezi dvěma entitami, skupinový chat, upozornění, chybová zpráva a v neposlední řadě zpráva bez kontextu vyžadující odpověď příjemce. Nakonec nesmí být opomenut blok zprávy, pro uživatele IM nejdůležitější, nesoucí vlastní obsah.

Základní použití struktury elementu *message* je prezentováno na příkladu 2.2. Na prvním řádku je uveden atribut, značící odesílatele. Druhý řádek obsahuje JID klienta, který zprávy přijímá. Následuje informace o typu zprávy a poté je uveden element *body* nesoucí samotný obsah.

---

```

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

```

---

Příklad 2.2: Použití elementu *message*.

Další částí stanzy je poskytována struktura pro *request-response* (žádost-odpověď) vazbu, podobnou metodám GET, POST a PUT z protokolu HTTP [25]. Zkráceně je označována



210 pomocí dvou počátečních písmen *Info/Query* neboli IQ. Na rozdíl od elementu *message*  
 211 tvoří *iq* spolehlivější přenos, optimalizovaný pro výměnu dat (binární data). K dalším roz-  
 212 dílům patří povinnost příjemce odpovědět na každou přijatou zprávu, neboli potvrdit její  
 213 doručení. Skutečnost, že je na právě požadovanou zprávu odpovězeno, zajišťuje parametr  
 214 *id*. Iq dotaz nebo odpověď musí obsahovat stejnou hodnotu tohoto atributu jako zpráva  
 215 vytvořená žádajícím subjektem. Další povinný atribut rozděluje iq na čtyři typy. Jednotlivé  
 216 žádosti na proces nebo akci jsou posílány samostatně [24]. V příloze C je uvedena rozsáhlejší  
 217 struktura tohoto elementu. Použití nachází v případech, které nastavují, žádají nebo infor-  
 218 mace posílají. Tato struktura je využívána pro novou registraci, posílání seznamu kontaktů  
 219 a další.

220 Příklad 2.3 znázorňuje základní použití elementu *iq*. Uživatel *user* posílá dotaz na získání  
 221 seznamu kontaktu (řádek 5.).

---

```

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

```

---

Příklad 2.3: Použití elementu *iq*.

222 Poslední a pro tuto práci nejdůležitější prvek stanzy je *presence*. V případě, že nemá  
 223 určeného příjemce, tak funguje způsobem jako broadcast. Což znamená, že jsou informace  
 224 směrovány všem klientům, kteří jsou zaregistrováni k jejímu odběru. Presence v českém  
 225 překladu informace o stavu (přítomnost) rozesílá dostupnost ostatních entit v síti. Jedná  
 226 se tedy o nastavení uživatelské dostupnosti tak jako na jiných real-time komunikačních  
 227 a sociálních systémech.

228 Existuje několik základních stavů statusů, které reprezentují aktuální dosažitelnost už-  
 229 vatele. Tento jev je vyjádřen pomocí elementu *show*, který disponuje čtyřmi možnostmi.  
 230 První oznamuje, že je uživatel k dispozici a schopen aktivní komunikace. Druhá často se  
 231 vyskytující možnost naznačuje, že je subjekt krátkou dobu pryč od svého IM klienta. Tento  
 232 a další dva stavy, popsané dále, jsou často změněny bez lidského zásahu (pomocí PC nebo  
 233 jiného zařízení) prostřednictvím funkce známé jako „auto-away“. Poslední dva stavy cha-  
 234 rakterizují delší časové období nečinnosti. Tato oznámení o změně stavu uživatele jsou často  
 235 zasílána pouze kontaktům, které se nacházejí v režimu online. Tato optimalizace přispívá  
 236 ke snížení síťového provozu, jelikož presence v reálném čase při komunikaci využívá velké  
 237 množství šířky pásma.

238 Základní použití *presence* je zobrazeno v příkladu 2.4. Kontakt *jabinfo@jabber.com/bot*  
 (1. řádek) posílá informace o svém stavu (řádek č. 2) a svůj status (č. 3).

---

```

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

```

---

Příklad 2.4: Použití elementu *presence*.

239  
 240 Obsáhlejší struktura elementu *presence* je zobrazena v příloze C, kde je rovněž k nalezení  
 241 přehled všech možných stavů.

242 Jak již bylo zmíněno v části o Jabber ID, Jabber podporuje práci s více současně připoje-  
243 nými klienty k jednomu Jabber účtu. Vysvětlení funkčnosti bude prezentováno na příkladu  
244 uživatele přihlášeného na stolním počítači a z klienta v mobilním telefonu. U obou těchto  
245 připojení je použit stejný Jabber bare, ale odlišného resource, například *domov* a *mobile*.  
246 Právě tento rozdíl v tzv. „full“ adrese účtu zajišťuje jednu ze dvou možných podmínek  
247 pro správnou adresaci zpráv. Druhá možnost, která bude uplatněna při použití adresy účty  
248 pouze ve formě Jabber bare, je nastavení priority u jednotlivých programů. Priorita je číslo  
249 v rozsahu hodnot od -128 do 127, kde klient s větší prioritou má přednost před klientem  
250 s nižší. Nastane-li případ připojení více klientů se stejnou prioritou, každý server se při ro-  
251 zesílání zpráv zachová podle vlastní implementace. Některé rozešlou zprávy všem klientům,  
252 jiné naopak jen poslednímu přihlášenému.

## 253 2.4 Rozšíření

254 Dále se tato technická zpráva zabývá rozšířeními protokolu XMPP o další vlastnosti, k je-  
255 jichž popisu slouží XEP. Pro tuto práci jsou nepostradatelné „statusy“, pro které tvoří zá-  
256 klad standardy XEP-0060 [12] a XEP-0163 [22] zkráceně PEP<sup>6</sup>. Obě tato rozšíření umožňují  
257 strukturovaně pracovat, používat a přenášet další XEP protokoly. Jako příklady relevantní  
258 k práci jsou zde uvedeny protokoly *User Location* (kde se uživatel právě nachází) [6], *User*  
259 *Tune* (co uživatel poslouchá za hudbu) [18], *User Mood* (aktuální nálada uživatele) [21]  
260 a *User Activity* (co uživatel právě dělá) [11]. Jsou to tedy protokoly založené na PEP,  
261 které vyžadují podporu nejen v klientech, ale i na straně serveru (zobrazuje tabulka 2.1).  
262 S touto informací úzce souvisí další protokol XEP-0115 [7], který umožňuje zjistit podporo-  
263 vané schopnosti klienta, případně, které informace je ochoten přijímat. Tato vlastnost bude  
264 popsána níže v části zabývající se podporovanými vlastnostmi.

265 Všechna tato rozšíření by mohla být přidána přímo do statusu viz příklad 2.4, avšak  
266 ten je primárně určen k informování o přítomnosti na IM síti. Hlavní rozdíl mezi PEP  
267 a obyčejným posílání stavu pomocí presence je v pravomoci klienta přijmout nebo odmítnout  
268 informaci, na rozdíl od presence, jež je přijata vždy.

269 Základ přenosu informací začíná na straně klienta, který chce všechny ve svém roster  
270 listu (seznam kontaktů), informovat o statusu. Zašle zprávu obalenou v elementu *iq* serveru.  
271 Ukázka této zprávy je prezentována na příkladu 2.5, který znázorňuje zaslání informace  
272 o druhu hudby, kterou v danou chvíli uživatel poslouchá. Využívá k tomu rozšíření *User*  
273 *Tune*, definovaném na řádku číslo 5. Základ zprávy oznamující začátek vysílání informací  
274 o rozšířených statusech je vždy stejný. Liší se pouze řádkem 3. a obsahem elementu *item*  
275 v příkladu 2.5.

---

```
1      <iq from='user@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                      ...
```

---

Příklad 2.5: Začátku vysílání rozšířeného statusu.

---

<sup>6</sup>Personal Eventing via Pubsub

276 V případě úspěšného přijetí *iq* zprávy serverem, každý, kdo se zaregistroval k odebrání  
277 rozšířených statusů, obdrží oznámení ve formě *message*. Oznámení bude také doručeno všem  
278 resources. Celá zpráva i všechny další náležitosti jsou uvedeny v příloze C.

## 279 Podporované vlastnosti

280 Jednotlivá rozšíření protokolu XMPP jsou nepovinná, a proto nemusí být ve všech klient-  
281 ských aplikacích podporována. Pro zjištění podporovaných rozšíření se používá XEP-0115  
282 Entity Capabilities [7]. Toto rozšíření výrazně snižuje počet a velikost komunikací a přenosů  
283 zpráv mezi uživateli. Dotazem zobrazeným na příkladu 2.6 je zjištěna schopnost jednotli-  
284 vých klientů, kterou následně server využije pro správné směřování rozšířených statusů.  
285 Všechny zde zmiňované rozšíření a protokoly z této kapitoly je možné u každého klienta  
286 (seznam klientů obsahuje tabulka v příloze C) vyčíst z atributu *ver* (druhá část u atributu  
287 node), který je vypočítán ze všech podporovaných protokolů klienta, viz [7].

---

```
1<iq from="user@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>
```

---

Příklad 2.6: Dotaz na podporované protokoly.

## 288 Další rozšíření

289 V následujících několika odstavcích budou přiblíženy specifikace jednotlivých rozšíření XEP,  
290 které slouží jako zdrojová data pro dolování a jsou relevantní k tématu práce.

291 Prvním rozšířením, nad rámec základních vlastností Jabberu, které zde bude podrobněji  
292 rozebráno, je elektronická verze klasické vizitky neboli *VCard*. Jeho specifikací se zabývají  
293 dva standardy. Jelikož novější verze XEP dokumentu [13] se v době psaní této práce nachá-  
294 zela ve stavu „experimental“, což znamená, že ještě není schválena jako standard, je pouze  
295 ve stavu návrhu. Proto bylo použito verze starší [16]. Jednoduše řečeno je *VCard* struktura,  
296 která nese informace o uživateli jako je jméno, příjmení, e-mail, adresa bydliště i zaměst-  
297 nání a další údaje. Data jsou dále zveřejňována na síti, z čehož vyplývá, že jsou dostupná  
298 ostatním uživatelům. Vyplnění těchto osobních údajů je dobrovolné a tak se u některých  
299 uživatelů nachází pouze přezdívka a JID, které jsou často předdefinovány automaticky. Ne-  
300 dílnou součástí všech sociálních a komunikačních systému jsou malé fotografie, loga nebo  
301 ikony, kterými se uživatelé prezentují. V síti Jabber tomu není jinak, a proto je samotný  
302 obrázek zahrnut přímo do *VCard* v položce *photo*. Podrobnější informace o jeho nastavení  
303 a přijímání je možné nalézt v *vCard-Based Avatars* [19], který jej definuje.

304 Díky základní podmínce XMPP protokolu (otevřenost) existuje mnoho různých apli-  
305 kací, pomocí kterých lze v síti Jabber komunikovat. S programy, používanými uživateli,  
306 úzce souvisí další zde implementované rozšíření. Jedná se o realizaci *Software Version* do-  
307 kumentu [17], který se právě zabývá získáváním informací o samotných aplikacích. Je-li  
308 toto rozšíření podporováno je díky němu možné zjistit jméno a verzi používané aplikace.  
309 Informace o operačním systému často nejsou kvůli bezpečnosti ani vyplněny. Podrobnější  
310 informace o softwarové výbavě klienta je možné zjistit pomocí XEP [7], o kterém již bylo  
311 dříve psáno v odstavci zabývajícím se podporovanými vlastnostmi klientských aplikací.

312 S rozšířením tzv. „chytrých“ mobilních zařízení mezi širší veřejnost vzniklo několik no-  
313 vých disciplín spojených s určováním zeměpisné polohy, jako je například geocaching. Geo-  
314 grafická poloha je přenášena ve formě souřadnic popisující přímo zeměpisnou šířku a délku.  
315 Současně lze informaci o poloze přenášet i slovně ve formě adresy. Příkladem slovního po-  
316 pisu je ulice, číslo popisné, město a další. Mnoho aplikací, které mají k dispozici GPS  
317 přijímač, vysílají a aktualizují zeměpisné informace automaticky, například po určité době  
318 nebo změně polohy o určitou vzdálenost. Toto a další níže popsání rozšíření jsou postaveny  
319 na již zmiňovaném PEP. Některé části protokolů jsou zjednodušeny a připraveny tím pro  
320 „mobilní instant messaging“.

321 Pro sdělení informací o stavu klienta není v základní verzi Jabberu mnoho. Pomocí  
322 presence je možné „pouze“ prozradit, zda je uživatel připraven komunikovat nebo je mo-  
323 mentálně nedostupný a to v několika verzích lišících se délkou nepřítomnosti. Pokročilejší  
324 nastavení statusu nabízí *User Mood* [21] a to ve formě sdělení současné nálady, jako je  
325 například radost. Další možné upřesnění činnosti uživatele jsou definovány v *User Activity*  
326 [11], kde každá činnost je složena z povinné obecné kategorie a nepovinné, která informaci  
327 upřesňuje. Příkladem může být *eating* a *having\_a\_snack* tj. uživatel jí, uživatel svačí.

328 K poslednímu rozšíření implementovanému v této práci patří *User Tune* [18], které  
329 umožňuje uživateli šířit informace o aktuálně poslouchané hudbě. Některé dnešní hudební  
330 přehrávače dokáží automaticky spolupracovat s IM klientem a předávat informace o hudbě  
331 bez nutného lidského zásahu. Ve zprávě jsou tedy přenášeny informace o skladbě, interpre-  
332 tovi, albu a další informace, které mohou být získávány z MP3 ID3v1 nebo novější ID3v2  
333 tag.

334 Podpora výše popsaných rozšíření v aplikacích je poměrně malá. Například v předchá-  
335 zející zmiňované části o poslouchané hudbě, při stavu, kdy program toto rozšíření nepod-  
336 poruje, je posíláno pomocí normální presence. Jméno skladatele, alba a další podrobnosti  
337 jsou shrnuty do statusu, tudíž jsou doručeny všem uživatelům ze seznamu kontaktů.

## 338 Kapitola 3

# 339 Data mining

340 Třetí kapitola se zabývá procesem dobývání znalostí z databází. Popisuje jej jako disciplínu,  
341 která vznikla za účelem vytěžení informací z dat, která jsou v nepřehledném množství uklá-  
342 dána v databázích. Díky velikosti dnešních disků, objem ukládaných dat neustále roste.  
343 S tím také úzce souvisí zvětšující se poměr nepotřebných a zašumělých dat vůči užitečným  
344 informacím. V této kapitole jsou mimo jiné popsány metody používané k dolování z dat,  
345 které jsou relevantní k této práci.

346 Na začátku kapitoly je rozebrán pojem získávání znalostí databází, jehož jednu podstat-  
347 nou část tvoří samotný data mining. Dále je vysvětlena základní terminologie, pro kterou  
348 bylo čerpáno z [8]. Cílem první podkapitoly je přiblížení způsobu, jakým byla data uložena  
349 v databázi a následně připravena k samotnému dolování z dat. Celý druhý oddíl je věno-  
350 ván přípravě dat pro samotné dolování z dat. Je zde popsán proces transformace atributů  
351 na proměnné kvantitativní. Poté následuje třetí podkapitola, která se podrobněji zabývá  
352 jednou z metod pro dolování dat a to *shlukováním*. Obsahem této části jsou již konkrétní  
353 algoritmy pro shlukování dat [29, 3] a také metoda *k-Means* využívaná v praktické části  
354 této práce. Kapitulu uzavírá stručný přehled vybraných programů pro data mining a po-  
355 drobnější seznámení s nástrojem *RapidMiner*, který je v této práci využíván pro samotné  
356 dolování.

## 357 Terminologie

358 Pojem data mining neboli česky dolování dat se začal ve vědeckých kruzích objevovat  
359 počátkem 90. let 20. století. První zmínka pochází z konferencí věnovaných umělé inteligenci  
360 (IJCAI'89<sup>1</sup> — mezinárodní konference konaná v Detroitu, AAAI'91<sup>2</sup> a AAAI'93 — americké  
361 konference v Kalifornii a Washingtonu, D. C) [2].

362 Tradiční metoda získání informací z dat je realizována jejich manuální analýzou a inter-  
363 pretací. V praxi ji například nalezneme v odvětví zdravotnictví, vědy, marketingu (efektivita  
364 reklamních kampaní, segmentace zákazníků) a dalších. Pro tyto a mnoho dalších disciplín  
365 je manuální zpracování příliš pomalé, drahé a vysoce subjektivní. Další důvod k přechodu  
366 na jiné metody je objemnost dat, která dramaticky vzrostla, a tudíž se manuální analýza  
367 stává zcela nepraktická. Databáze rychle rostou ve dvou následujících kategoriích:

- 368 1. počet záznamů neboli objektů v databázi
- 369 2. počet polí neboli atributů objektů v databázi

---

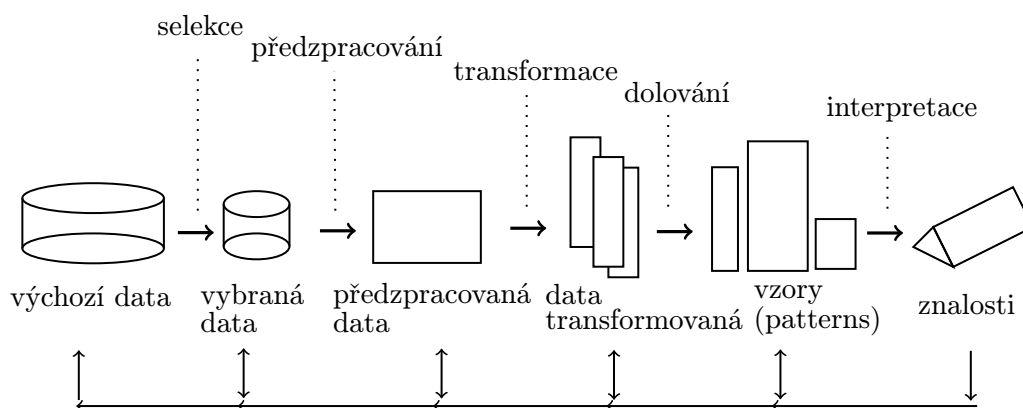
<sup>1</sup>International Joint Conference on Artificial Intelligence

<sup>2</sup>Association for the Advancement of Artificial Intelligence

Proces data mining je pouze jedna část z odvětví nazývané dobývání znalostí z databází neboli KDD<sup>3</sup> definované níže v definici 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, která má k uživateli nejblíže. Jako příklad bude uvedena 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, předzpracováním, 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 prezentuje časový harmonogram v KDD. Schéma znázorňuje následnost jednotlivých procesů, které tvoří KDD. KDD je iterativní proces, z čehož vyplývá, že skutečnosti nalezené v předchozích částech zjednoduší a zpřesní vstupy pro následující fáze. Jakmile jsou znalosti získány, jsou prezentovány 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 [4].

Vzhledem k obrázku 3.1, který prezentuje jednotlivé kroky získávání znalostí z databází, budou dále tyto procesy popsány. Prvním část v KDD je tvořena výchozími daty, které slouží jako zdroj pro ostatní fáze. Samotný popis získávání těchto dat je popsán v předcházející kapitole. Procesu selekce dat, je kladen za cíl, vybrat co možná „nejužitečnější“ množinu dat a tím i zmenšit její celkový objem. V této části získávání znalostí se při vybírání dat bere ohled na to, jak se jednotlivá data vztahují ke konkrétnímu uživateli. Následující proces nazvaný transformace se zabývá převedením dat do vhodného formátu pro samotné dolování informací. Tato část je popsána v následující podkapitole, kde hlavní úlohu při transformaci je čas. Vybrané metody pro dolování dat jako je například shlukování a další, jsou taktéž v této práci popsány níže.

<sup>3</sup>Knowledge Discovery in Database

397 Získávání znalostí z databází je proces složen z několika kroků vedoucích od surových  
398 dat k formě nových poznatků. Iterativní proces je složen, tak jak je prezentováno v [5],  
399 z následujících kroků:

- 400 • **čištění dat** — fáze, ve které jsou nepodstatné údaje odstraněny z kolekce.
- 401 • **integrace dat** — kombinování heterogenních dat z několika zdrojů do společného  
402 jediného zdroje.
- 403 • **výběr dat** — rozhodování o relevantních datech.
- 404 • **transformace dat** — také známý jako konsolidace dat. Fáze, ve které jsou vybraná  
405 data transformována do formy vhodné pro dolování.
- 406 • **data mining** — zásadní krok, ve kterém jsou aplikovány vzory na data.
- 407 • **hodnocení modelů** — vzory dat zastupují získané znalosti.
- 408 • **prezentace znalostí** — konečná fáze, zjištěné poznatky jsou reprezentovány uživa-  
409 teli. Tento základní krok využívá vizualizační techniky, které pomáhají uživa-  
410 telům porozumět a správně interpretovat získané výsledky.

411 Jak je uvedeno v [5], běžně jsou některé z těchto kroků kombinovány dohromady. Kroky  
412 čištění dat a integrace dat mohou být provedeny společně, tak jako to prezentuje schéma  
413 na obrázku 3.1.

414 V této podsekci jsou ve stručnosti vysvětleny základní nejdůležitější pojmy dále v práci  
415 využívané.

416 Definic výrazu data mining se v odborné literatuře nachází několik. Zde uvedená je  
417 kombinací dvou „definic“ z [15].

418 **Definice 3.0.2** Data Mining je proces objevování znalostí, který používá různé analytické  
419 nástroje sloužící k odhalení dříve neznámých vztahů a informací z velmi rozsáhlých databází.  
420 Výsledkem je predikční model, který je podkladem pro rozhodování [15].

## 421 3.1 Transformace dat

422 Transformace dat tvoří třetí část z celkového procesu dobývání znalostí z databází. Než se  
423 data dostala do tohoto stavu, bylo na nich provedeno několik kroků, ve kterých byla upra-  
424 vována. V první fázi byla sbírána Jabber komunikace, která je popsána v první kapitole.  
425 Druhá fáze byla zaměřena na zúžení výsledné množiny, a proto byla vybrána jen relevantní  
426 data. I přes tyto kroky relační databáze obsahuje velké množství dat, která se nenachází ve  
427 stavu, aby mohla být použita jako zdroj pro data mining. Jak bude popsáno v následující  
428 podkapitole většina metod pro dolování dat pracuje pouze s daty, která obsahují kvanti-  
429 tativní proměnné. Za tímto účelem je potřeba všechny atributy tabulky z databáze, které  
430 mají být nadále používány, převést na měřitelné hodnoty.

V této práci se bude pracovat s atributy nesoucí informace o jak aktuálních tak minulých  
stavech uživatelů, kteří si přidali účet *jabInfo@jabim.com* do svého seznamu kontaktů.  
A tak byla jejich každá změna statusu uložena do databáze. Z důvodu nečíselného hodnoty  
stavů, jako je například *available*, *away* a další, je třeba provést jejich transformaci na  
kvantitativní hodnoty. Proces byl proveden pomocí bijektivního zobrazení. Kde zobrazení



je podle [8] funkce s definičním oborem  $S$  a oborem hodnot  $T$ , která je nazývána binární relace  $f \subseteq S \times T$ . V této relaci se nevyskytují dvě různé dvojice  $(s, t_1)$  a  $(s, t_2)$ , kde  $s \in S$  a  $t_1, t_2 \in T$ . Prvky  $t_1$  a  $t_2$  jsou různé. Z toho vyplývá, že každému prvku  $s$  z množiny  $S$  je přiřazen jednoznačně právě jeden prvek  $t \in T$ . Tuto definici je možné zapsat ve tvaru:

$$f : S \rightarrow T,$$

431 kde  $S$  a  $T$  jsou množiny  $(D_f, H_f)$ .

Konkrétní případ transformace z této práce tedy bude obsahovat množinu  $S$ , kde

$$S = \{Available, Chat, Away, DND, XA, Unavailable\}$$

a množina  $T$ , kde

$$T = \{120, 110, 90, 70, 50, 0\},$$

432 do které budou jednotlivé prvky z množiny  $S$  bijektivně zobrazeny. Kdy bijekce je zobrazení,  
433 které každému prvku z cílové množiny, konkrétně z množiny  $T$ , přiřazuje právě jeden prvek  
434 z množiny počáteční, tedy  $S$ .

435 Při výběru velikosti hodnoty, pro výslednou množinu  $T$ , bylo čerpáno z programu Ga-  
436 jim, který je multiplatformní klient s velkou podporou standardů a rozšiřujících protokolů.  
437 Hodnoty v množině  $T$  byly zvoleny tak, aby měly sestupné uspořádání, a aby bylo možné je  
438 dobře mezi sebou porovnávat. Jsou-li vybrány dvě hodnoty například available a unavaila-  
439 ble, vzdálenost mezi nimi musí být větší než vzdálenost například u hodnot chat a away. Na  
440 druhou stranu prvky ze vstupní množiny  $S$  jsou striktně definovány podle Jabber standardu,  
441 který je popsán v RFC [23].

## 442 Temporální data

443 Druhá podstatná transformace, pro kterou bylo čerpáno z [26], se tak jako první nezabývá  
444 transformováním dat textových na data, jejichž obsah by byl tvořen kvantitativními pro-  
445 měnnými. V této části jsou řídká temporální data transformována na hustá. Pro následné  
446 vyhodnocení a data mining je potřeba řádkům z tabulky presence přidat konečné časové  
447 razítko, které by vymezilo interval doby platnosti těchto dat.

448 Jak již bylo uvedeno, hlavním rozdílem mezi temporálními databázemi a ostatními je  
449 schopnost uchovávat časové údaje. Své uplatnění nachází v odvětvích, kde je potřeba zpraco-  
450 vávat stará a zároveň nová data, například v oblastech medicíny, finančnictví, monitorování  
451 a dalších. Jednotlivé záznamy, které jsou závislé na čase, jsou v databázích ukládány jako  
452 samotné body, tedy diskrétně. Přestože v reálném světě je většina těchto údajů z pohledu  
453 času spojitých.

454 K dalšímu popisu temporálních databází nyní budou charakterizovány tři důležité po-  
455 jmy, pomocí nichž jsou databáze dále děleny. Prvním pojmem je *granualita*, která udává  
456 nejmenší časovou jednotku, kterou databáze rozlišují. Hodnoty, které může nabývat, jsou  
457 hodina, den, rok a další. Velikost granuality ovlivňuje velikost objemu dat, který je přímo  
458 úměrný s přesností záznamů. Dalším pojmem je *čas platnosti*, která reprezentuje období,  
459 kdy je daný fakt v modelovém světě pravdivý. Posledním termínem je *čas transakce*, která  
460 definuje přítomnost faktu v databázi a možnost jej získat. Čas transakce a čas platnosti jsou  
461 na sobě nezávislé a definují dvě rozdílné časové osy, kdy každá může disponovat s jinou gra-  
462 nualitou. Souhrnný název pro výše uvedené tři pojmy, který se používá, je systém časových  
463 razítek. Při použití těchto systému lze následně tvořit dotazy zaměřené na různá časová



464 období, jako je minulost, přítomnost a budoucnost. Tyto dotazy jsou velmi jednoduché a to  
465 díky rozšířenému jazyku TSQL, který vychází z klasické podoby dotazovacího jazyka SQL.

466 Temporální databáze jsou rozděleny, podle systému časových razítek, na tyto základní:  
467 *snímková, transakční, platného času, obojího času* (bitemporální). Entity z databáze, která  
468 je použita v této práci, je možné zařadit do kategorie *snímkových tabulek* (snapshot). K jejím  
469 hlavním rysům patří zaznamenávání stavu dat v jistém okamžiku. Čas transakce ani čas  
470 platnosti zde nejsou uplatněny.

471 Konkrétní příklad možné transformace je ukázán na části tabulky *presence* 3.1, která  
472 se ve stejném formátu nachází i v databázi, a modifikované tabulce *presence\_modify* 3.2.  
Tabulka 3.2 se od původní liší přidáním sloupce *dateEnd* (podbarven šedě), který s atri-

| id    | date              | toj               | presence  |
|-------|-------------------|-------------------|-----------|
| 87365 | 2011-4-4 13:53:59 | JabInfo@jabbim.cz | Available |
| 87369 | 2011-4-4 15:43:07 | JabInfo@jabbim.cz | Away      |
| ...   | ...               | ...               | ...       |

Tabulka 3.1: Ukázka tabulky *presence*.

473 butem *dateStart* vymezuje interval, kdy daná hodnota byla nebo je platná. Tato entita  
474 je pouze příkladem jak by mohla daná transformace vypadat. V této práci se žádná nová  
475 tabulka nevytvářela a ani se nemodifikovala již vytvořená. Celá transformace je popsána  
476 v další části této podkapitoly.

| id    | dateStart         | dateEnd           | toj               | presence  |
|-------|-------------------|-------------------|-------------------|-----------|
| 87365 | 2011-4-4 13:53:59 | 2011-4-4 15:43:07 | JabInfo@jabbim.cz | Available |
| 87369 | 2011-4-4 15:43:07 | INF               | JabInfo@jabbim.cz | Away      |
| ...   | ...               | ...               | ...               | ...       |

Tabulka 3.2: Ukázka modifikované tabulky *presence\_modify*.

477 Postup jednotlivých kroků při transformaci časových údajů z tabulky *presence*, je zob-  
478 razen v následujícím výčtu. Tento zjednodušený popis algoritmu vyžaduje dva vstupní  
479 parametry. První je JabberID uživatele, jehož položky v databázi mají být transformovány.  
480 Druhá nutná položka je datum, které bude sloužit jako upřesňující vstupní interval.

481 Data z tabulky *presence* jsou transformována na vektor  $\vartheta$  o 288 dimenzích, kde z pohledu  
časového je jedna dimenze období vymezené 5 minutami. Den je rozdělen na úseky po 5  
minutách ( $\delta = 300s$ ), kterých je 288. Tedy

$$\vartheta = (\vartheta_1, \vartheta_0, \dots, \vartheta_N),$$

482 kde  $N = 288$ .

- 483 1. Vstupním parametrem je datum, které vymezuje data pro transformaci. Toto datum je  
484 převedeno na dvě data (počátek a konec dne), která tvoří hraniční body v intervalu  $\iota$ .
- 485 2. Výběr dat z databáze, která splňují časové období definované intervalem  $\iota$  a uživatel  
486  $ID$ . Uložení těchto dat do množiny  $\Gamma$ .
- 487 3. Pokud zadanému dotazu neodpovídá žádný řádek z tabulky, jsou data označena jako  
488 prázdná. Výstupní transformovaný vektor  $\vartheta$  je naplněn hodnotami reprezentujícími stav  
489 *Unavailable*. Algoritmus je **ukončen**.

- 490 4. Zjištění prvního statusu toho dne.
- 491 4.1 Převod data o den dřívějšího na interval  $\iota_1$ , například  $\langle 2011-08-22\ 00:00:00, 2011-$
- 492  $08-22\ 23:59:59 \rangle$ .
- 493 4.2 Výběr dat vyhovujícím intervalu  $\iota_1$  a uživateli  $ID$  z databáze.
- 494 4.3 Výběr posledního záznamu z množiny dat získaných z předešlého dotazu.
- 495 4.4 Nalezení poslední presence a uložení její hodnoty do  $\lambda$ .
- 496 5. Výběr následujícího záznamu z množiny  $\Gamma$ .
- 497 6. Výpočet zda je časový interval mezi vybraným a následujícím záznamem větší jak  $\delta$ .
- 498 6.a Časový interval je větší než interval  $\delta$ .
- 499 6.1 Do výsledného vektoru  $\vartheta$  je ukládána hodnota presence daného záznamu.
- 500 6.2 Opakuj předešlý bod **6.1** kolikrát je interval  $\delta$  menší než rozdíl mezi časy
- 501 vybraného a následujícího záznamu.
- 502 6.b Časový interval je menší než interval  $\delta$ .
- 503 6.1 Jsou vybírány další záznamy z množiny  $\Gamma$  dokud rozdíl mezi časy v sekundách
- 504 není větší než interval  $\delta$ .
- 505 6.2 Jednotlivé presence záznamů jsou ukládány do pomocného pole, transformo-
- 506 vané do kvantitativních hodnot, jak je uvedeno v úvodu této podkapitoly.
- 507 6.3 Každý prvek pole je vynásoben počtem sekund, zastoupených v daném inter-
- 508 valu.
- 509 6.4 Výběr největšího prvku z pomocného pole a uložení jej do výsledného vektoru
- 510  $\vartheta$ .
- 511 7. Celý proces je opakován od bodu **5**, dokud množina  $\Gamma$  není prázdná.

## 512 3.2 Shlukování

513 Základ metod dolování dat je založen na statistice, posledních poznatcích z umělé inteli-

514 gence či strojového učení. Hlavní cíl těchto netriviálních metod je společný — snaha zjištěné

515 výsledky prezentovat srozumitelnou formou. Pro většinu používaných metod je společná

516 vlastnost předpoklad, že objekty popsané pomocí podobných charakteristik patří do stejné

517 skupiny (učení na základě podobnosti similarity-based learning). Objekty obsahující atri-

518 buty, lze převést na body v  $n$ -rozměrném prostoru, kde  $n$  reprezentuje počet atributů.

519 Vychází se z představy podobnosti bodů tvořící určité shluky v prostoru.

520 Další rozdíly mezi metodami, které byly prezentovány v [2], spočívají v:

- 521 • schopnosti reprezentace shluků (např. otázka lineární separability)
- 522 • srozumitelnosti nalezených znalostí pro uživatele (symbolické vs. subsymbolické me-
- 523 tody)
- 524 • efektivnosti znovupoužití nalezených znalostí
- 525 • vhodnosti typů dat

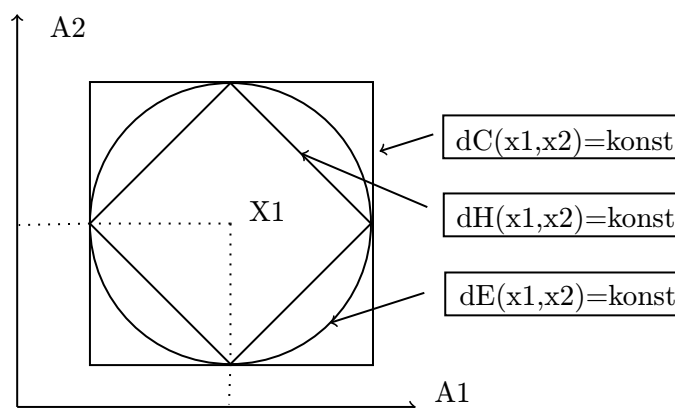
526 • a další ...

527 Problémy, které data mining řeší, se rozdělují do několika skupin. Do výčtu vybraných  
528 z nich, které budou následně rozebrány, patří *asociační pravidla*, *klasifikace*, *modely*, *pre-*  
529 *dikce* a *shlukování*.

530 Následující část se bude zabývat shlukováním, které je rozděleno na několik metod  
531 shlukové analýzy podle [5]. U každé z nich jsou popsány její základní vlastnosti a uvedeny  
532 algoritmy relevantní k práci. Poslední metoda *metoda rozkladu* je rozebrána podrobněji  
533 z důvodu jejího praktického využití v této práci.

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

536 Většina níže popsaných metod a algoritmů je založena na výpočtu vzdáleností mezi  
537 objekty. Tato vzdálenost lze vyjádřit různými mírami, podle knihy [2] například pomocí  
538 *Hammingovy vzdálenosti* (dH), *Euklidovské vzdálenosti* (dE) a *Čebyševovy vzdálenosti* (dC).  
539 Rozdíl mezi těmito typy určující vzdálenosti, graficky vyjadřuje obrázek 3.2. Kde  $X_1$  je  
540 střed, od něhož jsou jednotlivými obrazy znázorněny dané vzdálenosti. Konkrétně pomyslné  
541 body umístěné po obvodu kruhu jsou všechny stejně vzdáleny od středu  $X_1$ . Tato vzdálenost  
542 je označena jako Euklidovská. Další 2D těleso čtverec, který je vodorovný s osami  $A_1$  a  $A_2$   
543 prezentuje Čebyševovu vzdálenost. Po obvodu posledního obrazce, čtverce otočeného o  $45^\circ$   
544 podle osy  $A_1$ , jsou všechny pomyslné body stejně vzdáleny od bodu  $X_1$  Hammingovou  
545 vzdáleností.



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

546 Jako první jsou zde rozebrány *metody založené na modelu*, které se pokouší přiřadit  
547 data k určitému matematickému modelu na základě společných optimalizovaných vlastností.  
548 Většina procesů je založena na předpokladu, že jsou data generována pomocí standardních  
549 statistik. Mezi zástupné metody této shlukovací analýzy se řadí Expectation-Maximization  
550 (EM) a Self Organizing Oscillator Network, dále jen SOON. Algoritmus SOON je založen na  
551 neuronové síti. Je to metoda vycházející z algoritmu SOM<sup>4</sup> [29]. Metoda EM je rozšířením  
552 algoritmu *k-means*, který bude podrobně rozebrán v následující části.

553 Hlavní princip *metody hierarchického shlukování* je založen na tvorbě stromové hierar-  
554 chie shluků, která je známá pod názvem *dendrogram*. Hierarchické metody, podle [5], mohou  
555 být rozděleny do dvou skupin a to na základě principu, kterým jsou dendrogramy vytvářeny.

<sup>4</sup>Self-Organizing Map

556 První možnost je *aglomerativní přístup*, který shlukuje menší shluky, kdy výsledkem je jen  
557 jeden. Druhý přístup, *divizní*, je založen na opačném předpokladu. Na počátku je tedy jeden  
558 velký shluk, který je postupně rozdělován, dokud není počet shluků roven počtu objektů  
559 [29]. Mezi zástupce této metody například patří algoritmus AGNES<sup>5</sup>.

560 K dalším metodám patří *metody založené na mřížce*, které kvantují datový prostor do  
561 konečného počtu pravoúhlých buněk. Tyto buňky jsou uspořádány do víceúrovňové mříž-  
562 kové struktury. Zmíněná struktura tvoří základ pro shlukové operace. Hlavní výhoda tohoto  
563 přístupu je rychlost zpracování, které většinou nebere ohled na počet datových objektů. Čas  
564 zpracování závisí pouze na počtu buněk v každé dimenzi kvantovaného prostoru. Mezi zá-  
565 stupce metod založených na mřížce patří metoda STING — STatistical INformation Grid,  
566 který pracuje se statickými informacemi uloženými v buňkách mřížky. Algoritmus je roz-  
567 dělen do dvou částí. První si klade za cíl rekurzivně rozdělit datový prostor na pravoúhlé  
568 buňky. Druhá fáze testuje spojitost mezi sousedy relevantních buněk [29]. Mezi další metody  
569 založené na mřížce patří WaveCluster<sup>6</sup>, využívající vlnkové transformace k rozdělení pro-  
570 storu dat. Tato transformace zdůrazňuje shluky v prostoru a objekty jim vzdálené potlačuje  
571 [5].

572 *Metody založené na hustotě* vychází z  $m$ -rozměrného prostoru, ve kterém jsou zobrazeny  
573 objekty ve formě bodů. Místa v prostoru s větší koncentrací objektů ve srovnání s ostat-  
574 ními oblastmi jsou nazývány shluky. Výchozí předpoklad je existence okolí jednotlivých  
575 bodů (sousedství). Jedna z charakteristik metod založených na hustotě je schopnost vy-  
576 pořádat se s vzdálenými hodnotami, označovanými jako šum [29]. Jako příklad je uvedena  
577 metoda DBSCAN<sup>7</sup>, která je založena na hustotě objektů v prostoru. U jednotlivých ob-  
578 jektů je zkoumáno jejich okolí. Algoritmus je ovlivňován dvěma parametry, velikostí shluku  
579  $\varepsilon$  a minimálním počtem objektů v daném shluku  $MinPts$ , které spolu úzce souvisí (viz [5]).  
580 Bod splňující obě podmínky je označen za jádro. Za pomocí jader je rozšiřována množina  
581 objektů spojených na základě hustoty. Obsahuje-li jádro  $x_1$  ve svém okolí další centrum  
582  $x_2$ , znamená to, že  $x_1$  je přímo dosažitelné z  $x_2$ . Tímto způsobem jsou vytvářeny výsledné  
583 *shluky*. V opačném případě, body, které nesplňují dvě zmíněné podmínky, jsou označeny  
584 jako *šum*.

585 Všechny zde doposud zmiňované metody poskytují dobré výsledky pouze s malým poč-  
586 tem dimenzí, tak jak je to popsáno v [8]. S narůstajícím počtem atributů roste počet  
587 nerelevantních dimenzí určených pro shlukování. S tímto také přibývá zvětšená produkce  
588 zašumění a znesnadnění nalezení relevantních shluků. Data jsou roztroušena do mnoha di-  
589 menzí a tím odpadá možnost použití vzdálenostních funkcí. Zmíněné problémy shlukování  
590 velkých dat řeší dvě techniky *metoda transformace rysů* a *metoda výběru atributů*. Pro  
591 efektivní shlukování je možné použít například algoritmus CLIQUE<sup>8</sup>.

## 592 Metody rozkladu

593 Metody rozkladu rozdělují datové prvky do několika podmnožin, nazývané shluky. Počet  
594 shluků musí být znám před zahájením samotného procesu. Přiřazení do konkrétních tříd je  
595 podle [29], jednoznačné nebo probíhá na základě míry příslušnosti objektů do shluků. Pro  
596 velký počet objektů, se kterými se pracuje, jsou využívány různé iterační optimalizace.

597 Hlavním zástupcem u uvedených metod je algoritmus  $k$ -means, který je popsán níže.

---

<sup>5</sup>AGglomerative Nesting

<sup>6</sup>Clustering Using Wavelet Transformation

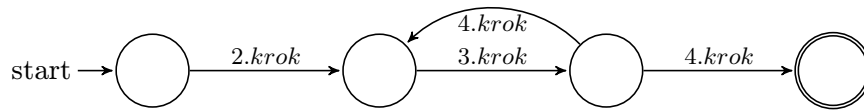
<sup>7</sup>Density-Based Spatial Clustering of Applications with Noise

<sup>8</sup>CLustering In QUEst

598 Tvoří základ pro většinu metod shlukování nejen pro metody rozkladu. K dalším metodám  
599 se řadí  $k$ -medoidů,  $k$ -modů,  $k$ -histogramů, fuzzy shluková analýza a další.

## 600 $k$ -means

601 Shlukování pomocí algoritmu  $k$ -means je používáno pro data obsahující kvantitativní pro-  
602 měnné a pro data, která nejsou příliš zašumělá. Základní proces je tvořen iterativním roz-  
603 dělováním objektů do tříd na základě vzdáleností od jejich středů. Střed neboli centroid  
604 shluku je vektor, jehož vzdálenost od součtu vzdáleností objektů v této třídě je minimální.  
605 Celý tento proces je prezentován na obrázku 3.3 pomocí jednoduchého schématu konečného  
606 automatu. Jednotlivé kroky konečného automatu odpovídají krokům  $k$ -means zobrazeného  
pomocí algoritmu 3.1. Pro výpočet vzdáleností mezi objekty samotnými nebo mezi objekty



Obrázek 3.3: Algoritmus  $k$ -means zobrazený pomocí konečného automatu.

607 a středem je použita euklidovská vzdálenost<sup>9</sup>, která je vyobrazena na obrázku 3.2.

609 K hlavním výhodám algoritmu  $k$ -means patří jeho relativní efektivnost. Složitost al-  
610 goritmu je  $O(TKN)$ , kde  $N$  je počet objektů,  $K$  je počet shluků a  $T$  je počet iterací.  
611 Obvykle platí, že počet objektů je mnohem větší než počet iterací i shluků. Na druhou  
612 stranu má i řadu nevýhod, kvůli kterým je často různými způsoby modifikován ( $k$ -medoids,  
613  $k$ -medians). K hlavním „nedostatkům“ patří předem nutná znalost počtu shluků (tříd)  
614  $K$ , do kterých budou objekty zařazeny. Druhý často se vyskytující problém je samotné  
615 ukončení algoritmu, které nastane u nalezení lokálního optima namísto optima globálního.  
616 Tato nepřesnost vzniká nevhodně zvoleným rozmístěním počátečních středů. Původní ne-  
617 modifikovaná verze algoritmu nedefinuje, jak se má postupovat, jsou-li nalezeny prázdné  
618 shluky.

619  $K$ -menas je algoritmus, kterým jsou přiřazovány objekty (vektory)  $x_n$ , kde  $n = 1, \dots, N$ ,  
620 do  $S_k$ , kde  $k = 1, \dots, K$ , shluků. V prvním kroku jsou určeny počáteční středy tříd, do  
621 kterých se budou objekty shlukovat. Určení počátečních centroidů  $c_k$  probíhá například  
622 náhodným výběrem  $K$  objektů nebo  $K$  prvních objektů souboru. Druhým krokem jsou  
623 zkoumány jednotlivé vzdálenosti objektů  $x_n$  od počátečních středů  $c_j$  pomocí euklidovské  
624 vzdálenosti. Na základě nejmenší zjištěné vzdálenosti mezi objektem a centroidem je ob-  
625 jekt zařazen do shluku, kterému náleží právě tento střed. Ve třetím kroku, tak jako u kroku  
626 prvního, jsou hledány nové středy shluků. Nyní již však nejsou zvoleny náhodně, ale spočí-  
627 tány. Jsou vypočítány na základě průměrných jednotlivých hodnot objektů a uložen jako  
628  $m$ -rozměrný vektor. Čtvrtým krokem se algoritmus dostává do konečné fáze, kdy mohou  
629 nastat dva možné případy. Nově nalezené středy nejsou příliš vzdáleny od předchozích  
630 centroidů a proto je algoritmus ukončen. Druhá častěji se vyskytující možnost iterativně  
631 provádí algoritmus od druhého kroku, dokud neplatí první možnost nebo dokud se objekty  
632 nepřestanou přemísťovat úplně. Při popisu tohoto algoritmu bylo čerpáno z [29, 2]. Níže  
633 zobrazený algoritmus 3.1 prezentuje krok po kroku metodu  $k$ -means.

<sup>9</sup>mean = střed, centroid je vektor průměrů

- 
1. náhodně zvol rozklad do  $K$  shluků
  2. urči centroidy pro všechny shluky v aktuálním rozkladu
  3. pro každý příklad  $x$ 
    - 3.1 urči vzdálenosti  $d(x, c_k)$ ,  $k = 1, \dots, K$ , kde  $c_k$  je centorid  $k$ -tého shluku
    - 3.2 nechť  $d(x, c_l) = \min_k d(x, c_k)$
    - 3.3 není-li  $x$  součástí shluku  $l$  ( $k$ -jehož centoridu  $c_l$  má nejblíže), přesuň  $x$  do shluku  $l$
  4. došlo-li k nějakému přesunu, potom jdi na 2, jinak konec
- 

Algoritmus 3.1: Metoda  $k$ -means byla převzata z [2].

634 Díky jednoduchosti a relativní rychlosti je metoda  $k$ -means stále výrazně využívána.  
 635 Uplatnění nachází v široké škále oblastí jako je například biologie nebo počítačová grafika.  
 636 Vzhledem k enormnímu počtu možného uspořádání nejsou výsledky vždy přesné, ale často  
 637 pouze přibližné.

### 638 3.3 Programy

639 V současné době na programovém trhu existuje mnoho systému, které jsou zaměřeny na data  
 640 mining. Mezi nejrozšířenější a nejdostupnější nástroje patří Weka a RapidMiner. K těmto  
 641 nástrojům je také možné zařadit program FIT-miner vyvíjený na fakultě informačních tech-  
 642 nologií v Brně. V této práci bylo pro samotné dolování z dat využito programu RapidMiner,  
 643 který dostal přednost před ostatními. Z pohledu nástroje FIT-miner, který ve své základní  
 644 části podporuje z databází pouze Oracle, se RapidMiner jevil jako vhodnější. Kompatibilitu  
 645 pro databáze typu PostgreSQL již měl zabudovanou a tak nebylo potřeba vyvíjet žádné do-  
 646 plňující moduly, jak by to bylo u FIT-mineru. V případě nástroje Weka, RapidMiner působil  
 647 propracovanějším dojmem a také nabízí lepší grafické zobrazení vyhodnocených výsledků.

648 Dalším velmi rozšířeným a často používaným nástrojem je jazyk  $R$ .  $R$  vychází z jazyka  
 649  $S$ , který ale není jako jazyk  $R$  volně šiřitelný. Statistický a grafický nástroj  $R$  je tedy volně  
 650 dostupným jazykem a prostředím, které je ovládáno pouze z příkazové řádky. Pro jednodušší  
 651 práci jej lze rozšířit o grafické rozhraní jako je RkWard nebo R Commander. Samotnou  
 652 aplikaci lze rozšířit o mnoho statistických doplňků, které jsou taktéž zdarma.

653 Mnoho programů pro dolování dat je založena na přístupu vizuálního programování.  
 654 Jedná se o proces, při kterém je uživatelem, za pomoci grafických prostředků, navržen  
 655 algoritmus a další postup práce. Jako příklad lze uvést poloprofesionální aplikaci *Orange*,  
 656 u které jsou nejdůležitější části psány pomocí C++ a rozšíření lze implementovat v jazyce  
 657 Python.

658 Na vybraných technicky zaměřených vysokých školách existují skupiny, které se zabývají  
 659 výzkumem a vývojem nástrojů pro data mining. Jako příklad lze uvést již dříve zmiňovaný  
 660 FIT-miner z fakulty informačních technologií v Brně nebo také projekt LISp-Miner z Vysoké  
 661 školy ekonomické v Praze. LISp-Miner je otevřený akademický systém určený pro výuku  
 662 a výzkum metod pro dobývání znalostí z databází.

### 663 RapidMiner

664 RapidMiner je, tak jak je popsán na oficiálních stránkách produktu [28], celosvětově nej-  
 665 používanější open-source systém pro dolování dat. Je možné jej používat jako samotnou  
 666 aplikaci nebo jej začlenit jako komponentu do vlastních výrobků v podobě knihovny pro ja-  
 667 zyk Java. Pro zájemce je nabízen také ve verzích pro firmy, které jsou rozdílné v poplatcích,

668 podpoře pro zákazníka, záruce a dalších balíčků služeb zajišťující celkovou komplexnost  
669 a spolehlivost produktu.

670 Jak již většina podobných aplikací, je v současné době implementován v jazyce Java,  
671 díky které nabízí flexibilní nejen grafické prostředí. K vybraným základním rysům toho  
672 nástroje, tak jak jsou prezentovány firmou *Rapid-i*, patří: výkonné, přesto intuitivní grafické  
673 uživatelské rozhraní pro návrh procesů, jednoduché řešení pro transformaci dat, kontrola  
674 výsledků již při samotném návrhu a další. Nástroj RapidMiner podporuje širokou škálu  
675 metod a algoritmů pro data mining. Mnoho algoritmů je implementováno přímo v aplikaci,  
676 ale také je použito metod z konkurenčního softwaru Weka. V základní verzi určené pro  
677 veřejnost je k nalezení přes 100 procesů k modelování. Jsou zde zastoupeny jak metody  
678 klasifikační a asociační, tak i metody shlukovací, z nichž lze jmenovat například DBSCAN,  
679  $k$ -medoids a hlavně  $k$ -means.

680 K dalším schopnostem RapidMineru je možnost spuštění jeho samotného pomocí gra-  
681 fického rozhraní nebo z příkazové řádky. Jak již bylo uvedeno dříve, je také možné jej  
682 použít jako knihovnu v jazyce Java. V této práci jsou použity první dvě možnosti. Pomocí  
683 grafického prostředí byl vytvořen experiment, otestována jeho funkčnost a následně pro  
684 jednotlivá shlukování použita šablona procesu, která byla volána z příkazové řádky. Tato  
685 možnost je k dispozici díky tomu, že jsou projekty v programu RapidMiner ukládány do  
686 čitelné a strukturované formy za pomoci značkovacího jazyka XML.



## 687 Kapitola 4

# 688 Implementace

689 Obsahem čtvrté kapitoly je popis praktické části této práce. Jsou zde charakterizovány  
690 jednotlivé prvky, které byly použity jak pro získání dat, tak pro jejich následné uložení.  
691 V první části je prezentována struktura architektury této práce. Její grafické znázornění je  
692 ukázáno na obrázku 4.1.

693 Cílem této práce je dolování dat z Jabberu. Jak již bylo dříve napsáno, Jabber je real-  
694 time síťová služba díky níž mohou její uživatelé komunikovat, informovat nebo sdílet svůj  
695 status s jinými uživateli. Celá tato vzájemná komunikace skrývá rozsáhlé množství informací  
696 o klientech dané sítě. Všechna tato navzájem vyměněná nebo poskytnutá data následně  
697 poslouží jako zdroj samotnému dolování. Pro jejich uskladnění je využita databáze, jejichž  
698 strukturální návrh prezentuje obrázek 4.2.

699 Třetí část této kapitoly je zaměřena na Jabber klienta neboli robota, který je imple-  
700 mentován v jazyce C++ pouze s konzolovým rozhraním. Robot v této práci hraje roli  
701 pasivního uživatele, který informace pouze přijímá. Ve vybraných případech dokáže uží-  
702 vatele ze svého seznamu kontaktů vyzvat k zaslání odpovědi s informacemi na odpověď,  
703 o kterou žádal. Struktura samotného robota je popsána níže a reprezentována obrázkem  
704 4.3. V části o Jabber robotovi jsou také uvedeny informace o knihovně *gloox*, kterou jsou  
705 zprostředkovány všechny náležitosti Jabber komunikace.

706 Poslední podkapitola této části je věnována případnému pokračování této práce.

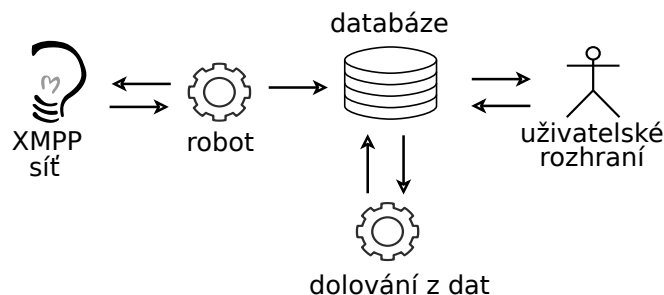
## 707 4.1 Architektura

708 Již ze samotného zadání a názvu této technické zprávy je zřejmé, že je třeba celý pro-  
709 ces rozčlenit na menší elementy. Vhodnou dekompozicí vzniklo pět jednotlivých ucelených  
710 prvků, které jsou prezentovány schématem na obrázku 4.1. Konkrétně to jsou, zleva: *XMPP*  
711 *server*, *robot*, *databáze*, *data mining* a *uživatelské rozhraní*. Vzájemná výměna informací  
712 mezi jednotlivými částmi je zobrazena pomocí šipek, které určují směr komunikace.

713 V levé části obrázku 4.1 je blok XMPP síť, který prezentuje zdroj dat. Fungování XMPP  
714 sítě je detailně popsáno ve druhé kapitole. Vztah a vzájemná komunikace s robotem pro-  
715 bíhá pomocí internetové sítě, kdy XMPP síť souhrnně reprezentuje jednotlivé prvky, jako  
716 jsou Jabber servery, uživatelé a jejich klienty. Jak je patrné, výměna informací mezi těmito  
717 částmi probíhá obousměrně. Druhý element schématu *robot*, je charakterizován v podkapi-  
718 tole níže, kde je podrobně popsán návrh jeho struktury. Jsou zde zdůrazněny jeho základní  
719 vlastnosti a schopnosti. Dalším blokem je *databáze*, která reprezentuje datové úložiště. Do  
720 něhož jsou za pomoci jednosměrného kanálu ukládána data, která jsou získávána z toku



721 informací proudícími mezi robotem a XMPP sítí. Přehled vybraných jednotlivých tabulek  
 722 a jejich atributů je možné nalézt v následující kapitole, která se zabývá návrhem databáze.  
 723 Dalším oddílem, prezentovaným na obrázku 4.1 pod názvem *data mining skript*, je  
 724 skript, který provádí samotné dolování z dat. Jako první jsou vybraná data z databáze  
 725 transformována do vhodného formátu pro dolování z dat. Tato část je podrobně popsána ve  
 726 třetí kapitole, v oddílu zabývajícím se transformací dat. Přeformátovaná data jsou předána  
 727 programu RapidMineru, kterým za pomoci algoritmu  $k$ -means je prováděn data mining.



Obrázek 4.1: Struktura architektury bakalářské práce.

728 Poslední částí je *uživatelské rozhraní*, které je implementováno pomocí webových služeb.  
 729 Jako příklad lze uvést službu, která zobrazuje status uživatele na webové stránce. Uživatel  
 730 pouze stačí vlastnit Jabber účet a mít přidáného tohoto robota do seznamu kontaktů. Od  
 731 každého uživatele, jež má tohoto robota přidáného do svého seznamů kontaktů, je sbírána  
 732 veškerá síťová aktivita. Možnosti, které mu nabízí uživatelské rozhraní v podobě webové  
 733 prezentace, tudíž záleží pouze na datech, které sám do sítě rozesílá. Jsou-li podporována  
 734 všechna zde implementována rozšíření, která jsou uvedena v kapitole druhé, je možné využít  
 735 jejich služeb prostřednictvím internetových stránek. Jako další příklad lze uvést zobrazení  
 736 aktuálního umístění uživatele na mapách od firmy google, dle informací poskytnutých pro-  
 737 střednictvím dokumentu User Geoloc [6].

## 738 4.2 Návrh databáze

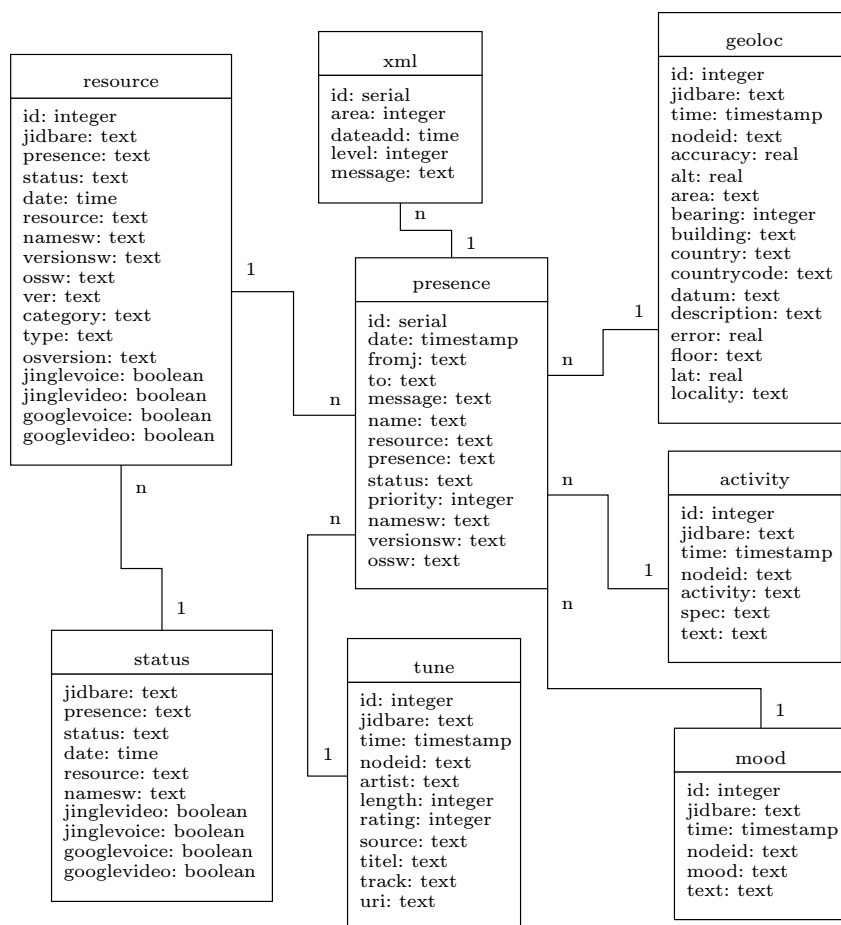
739 Data, která jsou sbírána robotem, jsou ukládána do objektově-relační databáze PostgreSQL.  
 740 PostgreSQL neboli také Postgres byl použit ve verzi 8.4.7 a je provozován na operačním  
 741 systému Ubuntu.

742 Struktura databáze, do které je ukládána veškerá komunikace Jabber robota, využívá re-  
 743 lační model. Obrázkem 4.2 jsou prezentovány nejdůležitější části databáze. Celou strukturu  
 744 návrhu je možné nalézt v příloze D. V jednotlivých částech návrhu databáze je počítáno  
 745 s druhotným využitím obsahu, které bude popsáno v dalších oddílech této práce.

746 V době návrhu databáze nebylo zcela zřejmé, která data budou následně analyzována.  
 747 Z tohoto důvodu se struktura databáze snaží zachytit všechna „důležitá“ data. Za tímto  
 748 účelem je v návrhu databáze obsažena tabulka *xml*, která je nositelem obsahu jak všech  
 749 přijatých, tak i odeslaných zpráv. Tabulky *debug*, *level* a *logarea*, které v zúženém návrhu  
 750 databáze, uvedeném na obrázku 4.2, nejsou zobrazeny, jsou určeny pouze jako doplňkové  
 751 informace k typu zprávy. Tabulku *xml* je možné nahradit jednotlivými dalšími tabulkami,  
 752 které jsou zaměřeny na konkrétní data. Příkladem entity, která již obsahuje konkrétní data  
 753 bez XML prvků, je tabulka *message*, která je taktéž vyobrazena v příloze C. Jejím obsahem

754 jsou zprávy vzniklé při Jabber komunikaci mezi uživatelem a robotem, jehož schopnosti  
755 budou popsány níže.

756 Tabulka *presence* z pohledu XMPP standardu prezentuje jeden ze tří základních částí  
757 stanzy, element *presence*. Základním cílem této tabulky je shromažďování uživatelských  
758 statusů. Jak již bylo zmíněno ve třetí kapitole, v části která se zabývá transformací, atri-  
759 but *presence* obsahuje hodnoty typu text. Příklad všech možných hodnot, kterých tento  
760 atribut může nabývat je prezentován v příloze C v části zabývající se elementem *presence*.  
761 K dalším atributům této tabulky patří *message*, který je reprezentován textovým řetězcem  
762 a rozšiřuje informace o stavu uživatele. V této části je často obsažen text, který je do sta-  
763 tusu přidán automaticky IM klientem. Transformovaný obsah této entity tvoří důležitou  
764 část při získávání znalostí a to v podobě dat, ze kterých budou „dolovány“ informace.



Obrázek 4.2: Vybraná část struktury databáze.

765 Většina rozšíření standardu XMPP, která jsou popsána ve druhé kapitole, jsou prezento-  
766 vána pomocí pěti tabulek. Konkrétně to jsou tabulky *geoloc*, *tune*, *mood*, *activity* a *vcard*.  
767 Pro obsah a názvy atributů všech pěti tabulek s rozšířeními se staly vzory dokumenty  
768 jednotlivých XEP, které je definují. Tabulky kromě těchto atributů obsahují také položku  
769 *jidbare*, která, jak již název naznačuje, obsahuje pouze čisté JabberID bez resources. Tato  
770 skutečnost vyplývá již ze samotného návrhu XEP protokolů. V tabulce *vcard* jsou také  
771 položky, jejichž obsah je tvořen fotografiemi. Obrázky jsou zde uloženy ve stavu, tak jak

jsou přenášeny po síti, tedy pomocí datového formátu Base64. Base64 je algoritmus, který převádí binární data na řetězec, který je tvořen pouze znaky ASCII tabulky.

S posledním rozšířením, nazvaném *Software Version* je spjata tabulka resource, jejíž několik sloupců tvoří informace o IM klientovi a operačním systému, na kterém běží. Tyto základní údaje jsou definovány v protokolu [17], kde je také možné čerpat bližší informace. Pokročilejší atributy, jako je *type* a *osversion* nacházejí svůj podklad v XEP dokumentu [7], taktéž jako atribut *ver*. V neposlední řadě se zde nachází zmínka o podpoře audia a videa, ať už v podobě *jingle* nebo *google*. Přesto tyto výše uvedené údaje nejsou hlavním důvodem existence této tabulky. Její primární cíl je uchovávat informace o připojených uživateli a všech jejich resources.

Entita resources tvoří základ pro tabulku *status*. Její obsah je automaticky generován z obsahu tabulky popsané výše. Počet řádků odpovídá počtu uživatelů, kteří jsou v seznamu kontaktů tohoto robota. Primárním cílem je informování o aktuálním stavu uživatele. Je tedy poskytován stav, ve kterém se uživatel nachází s přihlédnutím na prioritu a resources. Řádek entity obsahuje status a dostupnost uživatele a jeho resources s největší přihlášenou prioritou.

### 4.3 Návrh robota

V této části bude popsána aplikace, která zajišťuje real-time komunikaci s ostatními entitami Jabber sítě. Konzolový robot, který je vyvíjen pro operační systém Linux, také vhodným způsobem pracuje s databází, kam jsou ukládána jednotlivá získaná data. Jabber jádro síťové komunikace [23], je implementováno pomocí volně dostupné knihovny *gloox*. V porovnání s jinými knihovnami disponuje lepší podporou a dokumentací. Gloox plně implementuje standart XMPP Core [23] a z větší části i standard XMPP IM [24]. Dodatečně je podporováno kolem třiceti XEP standardů mezi něž například patří vcard-temp [16].

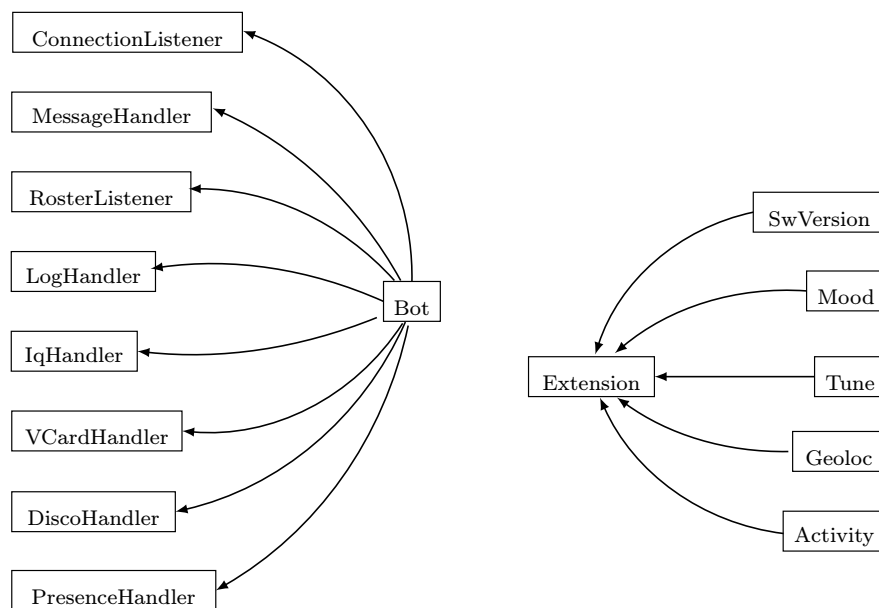
Při implementaci robota byly využity základní prvky objektově-orientovaného programování. Každá třída zde zastupuje jednoznačně oddělitelné elementy robota. Ať už se jedná o blok komunikující s databází, různá rozšíření nebo o jádro robota. Jako příklad lze uvést třídy, které reprezentují jednotlivá pro tuto práci vhodná rozšíření, jejichž základ je v podobě XEP dokumentů.

Struktura návrhu tříd robota, která je v základní části prezentována obrázkem 4.3, je velmi podobná struktuře návrhu databáze, obrázek 4.2. Mnoho atributů z robota a z databáze jsou ve vztahu jedna ku jedné. Samotná komunikace a řízení mezi těmito entitami je zprostředkováno pomocí knihovny *libpq*. V souboru *connect.h* jsou, v podobě předdefinovaných konstant, uvedeny všechny údaje nutné k navázání spojení ať už s databází nebo s komunikačním Jabber serverem.

Komunikace s databází se nachází v samostatné třídě, která zajišťuje provázanost mezi ní a daty. Pomocí funkcí jsou data, získaná z Jabber komunikace, přenášena a ukládána do jednotlivých tabulek databáze. Při zahájení činnosti robota je zajištěno navázání připojení k databázi a provedení nutných inicializačních kroků. Jako je tvorba nových tabulek, při prvním zapnutí aplikace, nebo kontrola existence těchto entit.

Pro rozšíření podle dokumentů XEP popsaných v kapitole druhé, v části zabývající se implementovanými rozšířeními, je základ nadtřída *Extension*. Tato třída obsahuje základní funkce a parametry pro všechna rozšíření. Třídy, které z *Extension* dědí a jsou prezentovány na návrhu robota uvedeném na obrázku 4.3, jsou následující: *Geoloc*, *Tune*, *Mood*, *Activity* a *SwVersion*. Poslední doposud nezmiňované rozšíření v podobě *vcard*, je využito z knihovny *gloox*, která jej implementuje.

818 Základní část, kterou je možné považovat za jádro aplikace, je třída *Bot*. Je implemen-  
819 tována pomocí tzv. „handler“ tříd z knihovny gloox, kterými je zajišťován přístup k jed-  
820 notlivým zprávám. Neboli jejich prostřednictvím je získán přístup k samotným elementům  
821 stanzy, jako je message, iq, a presence. Konkrétní seznam tříd, ze kterých je děděno, zob-  
822 razuje návrh robota na obrázku 4.3. Jako příklad lze uvést třídu, z prostoru jmen gloox,  
823 *VcardHandler*, díky níž je možné získat a zpracovávat vcard uživatelů, kteří jsou v se-  
824 znamu kontaktů robota. Samotné přidávání uživatelů do seznamu kontaktů je prováděno  
825 automaticky. Pouze na straně klienta je nutné požádat robota o autorizaci, která však pro-  
826 běhne automaticky. Jednotlivé kontakty nejsou žádným způsobem tříděny do skupin, tudíž  
827 existuje pouze jedna.



Obrázek 4.3: Vybraná část struktury robota.

828 K dalším schopnostem robota, tedy kromě sbírání dat a jejich následného ukládání do  
829 databáze, patří reagovat na vybrané zprávy. Tyto zprávy jsou přijímány pouze od uživa-  
830 telů, kteří byli začleněni do seznamu kontaktů tohoto robota. Základní příkaz, který nesmí  
831 chybět u žádné aplikace, je získání nápovědy. V první řadě informuje o verzi robota a jeho  
832 základních vlastnostech. Dále je odeslán stručný, ale výstižný seznam příkazů i s jejich vy-  
833 světlením. Při zaslání zprávy s neznámým příkazem robotovi, je uživateli vrácen stručný  
834 seznam kompatibilních příkazů. Přehled těchto příkazů se nachází v příloze E.

## 835 4.4 Pokračování práce

836 Obsah této podkapitoly je zaměřen na možnosti rozšíření robota sbírajícího data. Také jsou  
837 zde popsány možné změny, které se týkají uživatelského rozhraní. V závěru jsou shrnuta  
838 další rozšíření.

839 Většina služeb, které jsou dostupné na internetu a jsou podporované různými servery,  
840 poskytuje pouze informace o aktuálním stavu klienta. Pomocí výše uvedených standardů  
841 by bylo možné připojit k základní webové prezentaci statusu i informace rozšířené. Také za-  
842 pojení standardu *Entity Capabilities* [7], který je popsán na konci druhé kapitoly, by mohlo

843 přinést nové pokročilejší, užitečné informace. Konkrétně je to samotná podpora IM apli-  
844 kací v oblasti dalších XEP dokumentů. Nezasvěcený uživatel by tímto velice jednoduchým  
845 rozšířením dokázal zjistit seznam služeb podporovaných jeho IM programem. Ať už by se  
846 jednalo o seznam funkcí, které jsou pouze přijímány, nebo výčet rozšíření, k jejich odběru  
847 je zaregistrován. Jako příklad lze uvést zjištění schopnosti, ať už přijímání nebo odesílání  
848 informací o přehrávané hudbě, popsané dokumentem *User Tune* [18].

849 Další oblast, ve které lze nalézt možnost rozšíření, se také zabývá webovou prezentací.  
850 Nyní však jde již konkrétně o robota vytvořeného k účelu této práce. Informace potřebné  
851 k zobrazení aktuálního stavu klienta na internetové stránce, jsou uloženy v databázi v ta-  
852 bulce status. Tato tabulka je automaticky generována z obsahu jiné entity, která obsahuje  
853 údaje o všech připojeních daného uživatele. Vše pracuje správně až do té chvíle, kdy uživa-  
854 tel svou IM aplikaci neukončí korektně. Nastává situace, kdy na server není zaslána zpráva,  
855 která informuje o změně statusu. Díky tomuto neočekávanému ukončení klienta je uživatel  
856 prezentován, jako by byl připojen i když je tomu naopak. Zdrojová tabulka se v této chvíli  
857 stává neaktuální. Navrhované řešení této situace využívá rozšíření *XMPP ping* [20]. Díky  
858 němuž by se mělo dát zjistit, zda je uživatel stále dostupný nebo již nikoliv.

859 V oddílu robota by bylo možné pokračovat v části, která se zabývá příkazy zaslané ze  
860 strany uživatele. S tímto rozšíření úzce souvisí rozčlenění uživatelů do jednotlivých skupin.  
861 Každá takováto část seznamu kontaktů by vlastnila různá práva, která by se stala zákla-  
862 dem pro poskytování konkrétních služeb. Zařazení do těchto skupin by mohlo probíhat při  
863 žádosti o počáteční autorizaci, která by obsahovala zprávu s přesně definovaným řetězcem.  
864 Funkce, která je již uvedena výše a to v podobě rozšíření webové prezentace, by také mohla  
865 být poskytována prostřednictvím robota. Konkrétně se jedná o zjištění podporovaných ať  
866 již přijímaných nebo odesílaných rozšíření. Uživatel by pouze poslal robotovi striktně před-  
867 definovanou zprávu a jako odpověď by obdržel seznam podporovaných funkcí.

868 Také část zabývající se dolováním z dat skýtá velké množství možností pro její pokračo-  
869 vání. Jako jeden z příkladů lze uvést rozšíření založené na datech získaných prostřednictvím  
870 dokumentu *User Tune*. Konkrétně se jedná o internetové stránky, které by také mimo jiné  
871 nabízely prodej hudby. Jako vzor lze brát stránku *Last.fm* [27], která jednotlivým uživa-  
872 telům vytváří hudební profily na základě poslouchaných skladeb. Navrhované rozšíření by  
873 získávalo data pro tvorbu profilu z informací, které by byly šířeny prostřednictvím sítě  
874 Jabber a dokumentu *User Tune*. Využití dalšího standardu *User Location* by nahradilo  
875 manuální nastavení geografické polohy uživatele za automatické a tím by přineslo možnosti  
876 pro nabízení cílené reklamy.

877 S dokumentem *User Location* souvisí i další navrhované pokračování. Konkrétně se  
878 jedná o možný návrh nového XEP standardu, který by využíval určení geografické polohy  
879 například k naplánování cesty. Rozšíření by umožňovalo konkrétním lokalizačním souřadni-  
880 cím přiřazovat důležité body, informace a poznámky. Ty by byly ihned rozeslány způsobem,  
881 jakým pracují dosavadní rozšíření postavená na *Personal Eventing Protocol* [12]. Využití  
882 by nalezlo ať už v záchranných misích nebo jen k šíření zajímavostí mezi uživateli.

883 V měsíci březen, tohoto roku, vyšla nová verze základních dokumentů RFC. V této  
884 práci, již ale z těchto standardů nejsou zahrnuty žádné změny. V době vydání se totiž práce  
885 dostával do konečné fáze vývoje. Proto změny přinesené novými protokoly by taktéž mohly  
886 být uplatněny při případném pokračování této práce.

## 887 Kapitola 5

# 888 Vyhodnocení výsledků

889 Pátá kapitola je zaměřena na vyhodnocení výsledků a prezentaci zjištěných informací.  
890 V první podkapitole jsou uvedeny poznatky zjištěné z ručního prozkoumávání získaných  
891 dat, která jsou uložena v databázi. Z velké části je zde popsán jak vznik nepřesností, tak  
892 i jejich případné následné odstranění. V závěru této podkapitoly jsou shrnuta kvantitativní  
893 fakta zaměřená na databázi, jako je například počet záznamů a velikost samotné databáze.

894 Druhá podkapitola se zabývá samotným procesem dolování z dat, který je prováděn  
895 pomocí nástroje RapidMiner [28]. Je postavena na informacích získaných z druhé a třetí  
896 kapitoly. Prostřednictvím tabulky 5.2, jejíž celá podoba se nachází v příloze F, jsou pre-  
897 zentovány zjištěné informace. Jedná se o procentuální shodu uživatelů mezi sebou, která je  
898 založena na datech získaných z jejich statusů v Jabber síti.

## 899 5.1 Manuální rozbor dat

900 Data, která jsou získaná z Jabber komunikace, jsou uložena v databázi PostgreSQL. V průběhu  
901 celého vývoje této práce byl obsah databáze důkladně kontrolován a případné problémy  
902 řešeny v nejbližším možném termínu. I tak se ale v databázi nacházelo několik chyb, které  
903 při následné analýze dokázaly mírně znepřesnit výsledky. Jako příklad lze uvést zazname-  
904 návání rozšířeného statusu o geografické poloze do tabulky geoloc. V případě, že uživatel  
905 současně používal pouze jedno připojení, bylo vše v pořádku. Menší kolize nastala v situaci,  
906 kdy bylo více současných připojení. Každé rozesílalo své informace, které byly stejné, a tím  
907 se v jeden čas v databázi objevila redundantní data. Tato nepřesnost byla po manuálním  
908 rozboru dat objevena a odstraněna pomocí časových razítek.

909 Největší problém při sbírání dat, je tvořen díky aplikacím, které jsou využívány uživa-  
910 teli. Jejich neúplná implementace XEP dokumentů a vzájemná nepřesná spolupráce s ji-  
911 nými programy tvoří velké množství matoucích dat, které se taktéž nacházejí v databázi.  
912 Tento problém se v největší míře objevuje v tabulce tune, jejímž cílem je uchování informací  
913 o hudbě, kterou uživatelé poslouchají. Podle dokumentu User Tune [18], by po skončení pře-  
914 hrávání hudby měl klient rozeslat do „světa“ zprávu, jejíž obsah, nesoucí informace o právě  
915 přehrávané hudbě, bude prázdný. Tím je sděleno ukončení přehrávání hudby. V mnoha  
916 případech se, ale takto neděje. Zprávu, která obsahuje informace o poslední přehrávané  
917 hudbě, uživatel rozešle jako poslední. Tudíž z analýzy dat obsažených v databázi je mylně  
918 předpokládáno, že klient hudbu stále poslouchá.

919 Samotný sběr informací započal již v minulém roce v měsíci prosinec. V této době  
920 však ještě nebyla implementována všechna rozšíření a formát i obsah vybraných tabulek



byl později modifikován. Základní informace o uživatelském stavu, které jsou uchovávány v tabulce presence, již ale výraznou změnou neprošly a tudíž jsou sbírány od samého počátku běhu aplikace. V obsahu této entity pouze nejsou zahrnuty vybrané dny, kdy byl klient mimo provoz.

Databáze uchovává informace pouze o počtu kolem 40 uživatelů. Tento fakt výrazně ovlivnil přesnost a rozsah nasbíraných dat. Taktéž malé množství podpory rozšířených statusů ze strany uživatelů a jejich aplikací, značně ovlivnilo obsah databáze. Ať už do počtu záznamů nebo umístění informací do správných entit. S tímto taktéž úzce souvisí rozšíření User Tune, jehož nesprávné použití je popsáno v závěrečné části druhé kapitoly.

Za dobu, která přesahuje 4 měsíce, bylo nasbíráno téměř 200 MB dat. Například tabulka, která uchovává data pro data mining popsány v následující podkapitole, obsahuje pře 115 tisíc záznamů. Na druhou stranu entity zaměřené na rozšířené statusy, konkrétně to jsou geoloc, tune, mood a activity, disponují s menším počtem záznamů, který se počítá pouze ve stovkách. Tento omezený počet, jak již bylo uvedeno, je výsledkem menší podpory těchto rozšíření ze strany klientů.

Tabulka 5.1 souhrnně prezentuje nasbíraná data v jednotlivých kategoriích rozšíření. Prvními zjištěnými údaji je tvořen druhý sloupec, kterým je shrnut počet uživatelů podporujících daný standard. Jak druhý tak i třetí sloupec se skládá ze dvou čísel, která jsou vzájemně oddělena znakem lomítka /. První údaj uvádí počet užitečných záznamů z celkového počtu. Konkrétně v druhém sloupci to je počet uživatelů, kteří šířili užitečné informace, k počtu uživatelů, jež šířili jakákoliv data, tedy i prázdná. Tabulku uzavírá přehled průměrného počtu záznamů na jednoho uživatele.

| Rozšíření     | Uživatelé/z      | Záznamy/z | Průměr |
|---------------|------------------|-----------|--------|
| User Tune     | 10 %(4)/44 %(18) | 2258/4789 | 564    |
| User Location | 12 %(5)/12 %(5)  | 122/276   | 24     |
| User Mood     | 20 %(8)/31 %(13) | 550/1367  | 63     |
| User Activity | 10 %(4)/20 %(8)  | 404/1136  | 101    |

Tabulka 5.1: Přehled manuálního rozboru dat.

Manuální rozbor dat probíhal i při tvorbě webové prezentace, která se zaměřuje na praktické využití získaných dat. Lze ji nalézt na stránce <http://pcpanel-1205.fit.vutbr.cz/jabinfo/> na jejíž tvorbě se z velké části podílel vedoucí této práce Ing. Jozef Mlích. Obsah této ukázky slouží i jako názorný manuál, jak by se s daty mohlo dále nakládat.

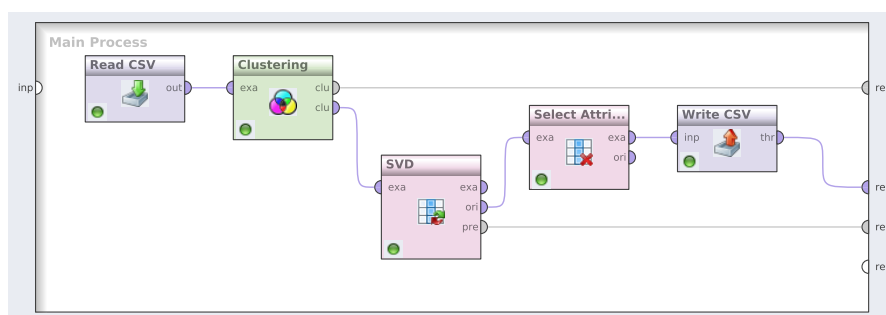
Následující podkapitola se zaměřuje na vyhodnocení údajů z tabulky presence. Toto rozhodnutí je postaveno na ručním prozkoumání získaných dat, které je shrnuto v tabulce 5.1. Jsou-li použita data z entit mood a activity je na první pohled patrné, že uživatelé využívající tyto služby, je ve své aplikaci nastavili pouze jedenkrát. Jelikož nastavení těchto rozšíření neprobíhá automaticky, nelze získané informace považovat za věrohodné. Pro samotnou analýzu by byl potřebný mnohonásobně větší objem dat, než jaký by byl nutný například pro tabulky tune a geoloc. Po prozkoumání tabulky geoloc bylo vyvozeno, že získané geografické polohy od klientů nejsou dále použitelné.

## 5.2 Programové vyhodnocení

V této části vyhodnocení jsou využita data získaná robotem z Jabber komunikace, která jsou uschována v databázi. Před samotným procesem dolování z dat, na řadu přichází

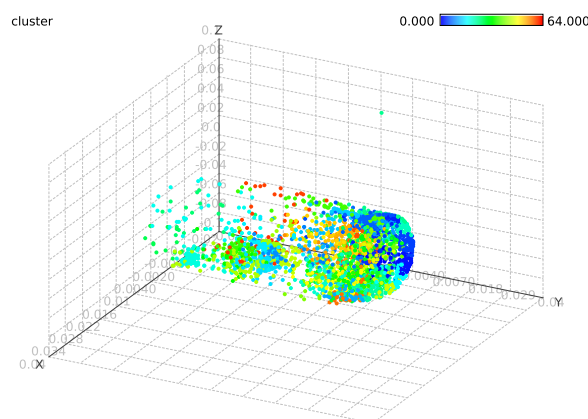
transformace popsaná v kapitole třetí, která data připraví. Je vytvořen formát, který je kompatibilní s programem RapidMiner [28]. Jak bylo uvedeno na konci předešlé kapitoly, rozšířené statusy pro proces dolování z dat nejsou vhodné. I z tohoto důvodu, ale hlavně díky velkému množství nasbíraných dat byla pro data mining využita tabulka presence, jejíž obsah je tvořen informacemi o změně statusů jednotlivých uživatelů.

Struktura samotného procesu vytvořeného v programu RapidMiner je zobrazena na obrázku 5.1. Jak je patrné je tvořena z několika bloků, kde každý plní určitou funkci. Vzájemné vztahy mezi jednotlivými oddíly procesu jsou reprezentovány úsečkami. První částí, která je nazvána *Read CSV*, jsou transformovaná data načtena z externího souboru do nástroje. Následující oddíl *Clustering* využívá algoritmus *k*-means, kterým se podrobně zabývá kapitola třetí. Počet shluků byl experimentováním nastaven na konečnou hodnotu 100. Blok nazvaný *SVD*<sup>1</sup> slouží ke snížení dimenzí jednotlivých vstupních vektorů. Pro



Obrázek 5.1: Struktura procesu v nástroji RapidMiner.

vizuální zobrazení bylo zvoleno snížení dimenzí z 288 až na hodnotu 3. Následujícím oddílem *Select Attribute*, který využívá data z bloku *Clustering*, jsou vybrány pouze dva atributy. Konkrétně to jsou jednotlivá JID uživatelů a jména shluků, do kterých byly zařazeny. Posledním blokem jsou tato data exportována do CSV souboru.



Obrázek 5.2: Výsledné shluky převedeny do 3D prostoru.

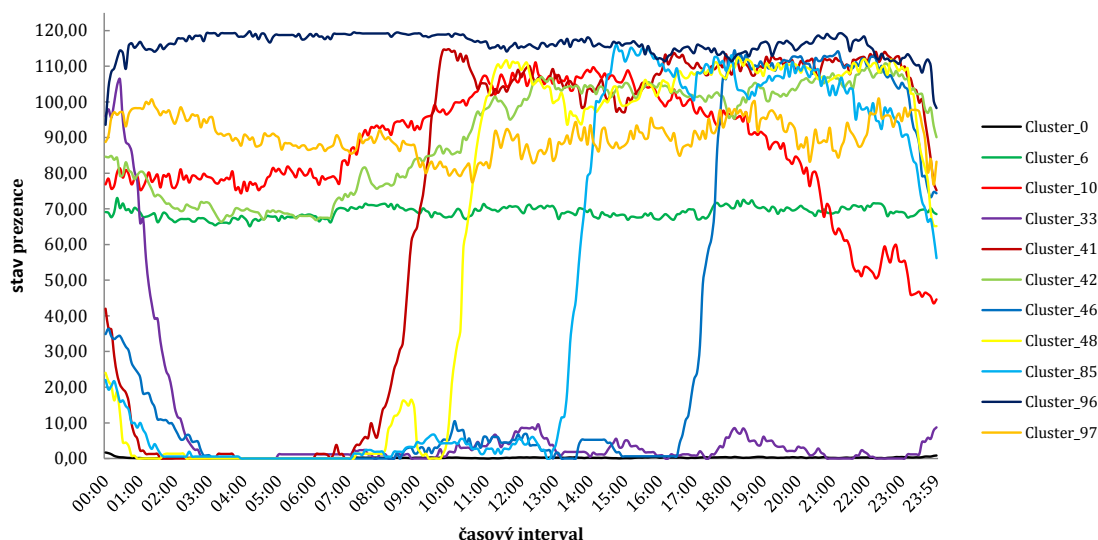
Obrázkem 5.2 je prezentováno rozmístění jednotlivých shluků v tří rozměrném prostoru.

<sup>1</sup>Singular Value Decomposition



975 Byl vygenerován automaticky blokem SVD zobrazeném na obrázku 5.1. Tento blok, jak již  
 976 bylo uvedeno, redukuje  $n$ -rozměrný vstupní vektor do tří dimenzí. Snižování prostorového  
 977 řádu bylo provedeno za účelem grafického znázornění jednotlivých shluků. Tyto výsledné  
 978 množiny jsou od sebe navzájem odlišeny barvami.

979 Pomocí algoritmu  $k$ -means byli uživatelé rozčleněni podle svého chování do několika  
 980 množin. Pouze jedenáct shluků obsahovalo více jak osmdesát záznamů a právě ty jsou pre-  
 981 zentovány na obrázku 5.3. Je zde znázorněn graf znázorňující vztah mezi statusem a časem.  
 982 Osa  $x$  je tvořena časovým intervalem v rozmezí jednoho dne a na ose  $y$  je vynesena aktuální  
 983 stav neboli presence.



Obrázek 5.3: Průběh statusů shluků podle času.

984 Po provedení shlukování algoritmem  $k$ -means pomocí nástroje RapidMiner jsou uživa-  
 985 telé rozřazeni do shluků a uloženi do externího souboru. V této chvíli se celkové vyhodnocení  
 986 dostává do závěrečné fáze. Na řadu přichází aplikace, pomocí které je vypočítáno zastoupení  
 987 jednotlivých uživatelů ve shlucích.

988 Vzájemná procentuální shoda uživatelů mezi sebou je prezentována tabulkou 5.2. Kde  
 989 hlavička, která je podbarvena šedě a umístěna na prvním řádku, znázorňuje vybrané uživa-  
 990 tele. Jejich samotné JID je, za účelem úspory místa a zachování soukromí, nahrazeno čísly  
 991 od jedné po dvaadvacet. Tato struktura je uvedena i v prvním sloupci. Je tedy vytvořena  
 992 „matice“, jejíž hlavní diagonála není vyplněna. Obsah je tvořen procentuální shodou mezi  
 993 jednotlivými uživateli. Buňky prezentující procentuální zastoupení větší než 80 % jsou ba-  
 994 revně odlišeny. Zeleně zabarvená políčka znázorňují podobnost v intervalu od 80 do 89  
 995 procent a červená barva značí interval shody mezi 90 až 99.

996 Výše zmíněná tabulka 5.2 prezentuje pouze vybranou část uživatelů. Celý přehled je  
 997 umístěn v příloze F v tabulce F.2. Jednotlivé nahrazení JID uživatelů za čísla je taktéž  
 998 uvedeno v příloze F, konkrétně v tabulce F.1.

999 Údaje prezentované tabulkou 5.2 byly i manuálně zkontrolovány a to vedlo k závěru, že  
 1000 celý proces počínaje transformací až po samotné vyhodnocení je navržen správně. Je patrné,  
 1001 že ke 100 procentní shodě nedošlo ani u uživatelů, kteří využívají dva stejné účty. Tento  
 1002 důsledek je možné vysvětlit tím, že ne vždy byly oba účty ve stejném stavu a ve stejné chvíli

| -  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1  | -  | 22 | 31 | 70 | 23 | 28 | 31 | 37 | 55 | 54 | 64 | 14 | 52 | 32 | 34 | 54 | 44 | 27 | 23 | 19 | 29 | 23 |
| 2  | 22 | -  | 63 | 24 | 90 | 88 | 84 | 55 | 52 | 39 | 34 | 90 | 29 | 14 | 56 | 32 | 44 | 89 | 85 | 50 | 33 | 22 |
| 3  | 31 | 63 | -  | 26 | 63 | 66 | 64 | 57 | 55 | 47 | 42 | 60 | 45 | 22 | 79 | 34 | 44 | 66 | 63 | 57 | 36 | 38 |
| 4  | 70 | 24 | 26 | -  | 23 | 29 | 30 | 36 | 53 | 48 | 58 | 14 | 42 | 26 | 28 | 58 | 41 | 26 | 21 | 18 | 30 | 19 |
| 5  | 23 | 90 | 63 | 23 | -  | 90 | 85 | 60 | 49 | 42 | 34 | 88 | 32 | 17 | 55 | 30 | 42 | 86 | 86 | 49 | 34 | 23 |
| 6  | 28 | 88 | 66 | 29 | 90 | -  | 86 | 60 | 56 | 44 | 38 | 85 | 34 | 17 | 58 | 36 | 42 | 87 | 85 | 49 | 32 | 24 |
| 7  | 31 | 84 | 64 | 30 | 85 | 86 | -  | 59 | 53 | 49 | 43 | 81 | 31 | 22 | 56 | 32 | 42 | 86 | 83 | 51 | 34 | 25 |
| 8  | 37 | 55 | 57 | 36 | 60 | 60 | 59 | -  | 60 | 53 | 48 | 51 | 41 | 21 | 60 | 39 | 51 | 58 | 59 | 50 | 43 | 24 |
| 9  | 55 | 52 | 55 | 53 | 49 | 56 | 53 | 60 | -  | 58 | 58 | 43 | 53 | 26 | 55 | 52 | 63 | 56 | 52 | 47 | 43 | 26 |
| 10 | 54 | 39 | 47 | 48 | 42 | 44 | 49 | 53 | 58 | -  | 60 | 33 | 49 | 31 | 48 | 55 | 56 | 44 | 42 | 39 | 44 | 29 |
| 11 | 64 | 34 | 42 | 58 | 34 | 38 | 43 | 48 | 58 | 60 | -  | 26 | 48 | 35 | 44 | 55 | 46 | 38 | 34 | 31 | 44 | 29 |
| 12 | 14 | 90 | 60 | 14 | 88 | 85 | 81 | 51 | 43 | 33 | 26 | -  | 24 | 9  | 51 | 25 | 36 | 82 | 82 | 49 | 29 | 21 |
| 13 | 52 | 29 | 45 | 42 | 32 | 34 | 31 | 41 | 53 | 49 | 48 | 24 | -  | 29 | 45 | 56 | 53 | 37 | 29 | 29 | 40 | 30 |
| 14 | 32 | 14 | 22 | 26 | 17 | 17 | 22 | 21 | 26 | 31 | 35 | 9  | 29 | -  | 24 | 25 | 22 | 17 | 14 | 16 | 23 | 19 |
| 15 | 34 | 56 | 79 | 28 | 55 | 58 | 56 | 60 | 55 | 48 | 44 | 51 | 45 | 24 | -  | 37 | 49 | 59 | 56 | 59 | 39 | 44 |
| 16 | 54 | 32 | 34 | 58 | 30 | 36 | 32 | 39 | 52 | 55 | 55 | 25 | 56 | 25 | 37 | -  | 49 | 34 | 30 | 27 | 36 | 25 |
| 17 | 44 | 44 | 44 | 41 | 42 | 42 | 42 | 51 | 63 | 56 | 46 | 36 | 53 | 22 | 49 | 49 | -  | 49 | 46 | 38 | 49 | 25 |
| 18 | 27 | 89 | 66 | 26 | 86 | 87 | 86 | 58 | 56 | 44 | 38 | 82 | 37 | 17 | 59 | 34 | 49 | -  | 86 | 51 | 36 | 23 |
| 19 | 23 | 85 | 63 | 21 | 86 | 85 | 83 | 59 | 52 | 42 | 34 | 82 | 29 | 14 | 56 | 30 | 46 | 86 | -  | 51 | 36 | 25 |
| 20 | 19 | 50 | 57 | 18 | 49 | 49 | 51 | 50 | 47 | 39 | 31 | 49 | 29 | 16 | 59 | 27 | 38 | 51 | 51 | -  | 34 | 35 |
| 21 | 29 | 33 | 36 | 30 | 34 | 32 | 34 | 43 | 43 | 44 | 44 | 29 | 40 | 23 | 39 | 36 | 49 | 36 | 36 | 34 | -  | 26 |
| 22 | 23 | 22 | 38 | 19 | 23 | 24 | 25 | 24 | 26 | 29 | 29 | 21 | 30 | 19 | 44 | 25 | 25 | 23 | 25 | 35 | 26 | -  |

Tabulka 5.2: Procentuální shoda jednotlivých uživatelů.

1003 zapínány. Taktéž se dospělo ke zjištění, které dokazuje, že ke správnému vyhodnocení by  
1004 bylo potřeba, aby všichni uživatelé měli přibližně stejný počet záznamů obsahujících změnu  
1005 statusu. K velké procentuální shodě například došlo u dvou klientů, kteří měli v databázi  
1006 pouze tři záznamy, což nemá žádnou vypovídající hodnotu o jejich podobnosti.

## 1007 Kapitola 6

## 1008 Závěr

1009 Cílem této bakalářské práce bylo vytvoření klienta pro real-time síť Jabber. Tato aplikace  
1010 pracuje jako robot, který je schopen reagovat na vhodně zvolené příkazy a sbírat data  
1011 z komunikace. Následujícím krokem bylo analyzování nashromážděných dat, se kterými byl  
1012 dále proveden proces dolování. Výsledky jsou prezentovány pomocí tabulky, jejíž obsah je  
1013 tvořen procentuální shodou jednotlivých uživatelů mezi sebou.

1014 Po důkladném vypracování této práce je možné prohlásit, že všechny cíle kladené na  
1015 tuto práci se podařilo splnit. Každému bodu zadání je zde věnován oddíl, ve kterém je  
1016 daná problematika podrobně rozebrána. Prvnímu bodu zadání odpovídá celá druhá kapi-  
1017 tola, která je tvořena popisem protokolu XMPP, na kterém je postavena komunikační služba  
1018 Jabber. Také je zde rozebrána architektura sítě a základní stavební kameny XMPP proto-  
1019 kolu, které jsou využívány sítí Jabber jako nástroj ke zprostředkování jednotlivých služeb.  
1020 Druhý bod zadání úzce souvisí s tvorbou robota, který byl vytvořen za účelem získávání dat  
1021 z Jabber komunikace. Zdrojové soubory této aplikace jsou přiloženy na kompaktním disku.  
1022 Implementační částí se věnuje kapitola čtvrtá, kde je rozebrána jak celková programová  
1023 architektura této práce tak její jednotlivé bloky. Prvním oddílem kapitoly páté je splnění  
1024 třetí bod zadání. Jsou zde uvedeny poznatky, které byly získány z manuálního průzkumu  
1025 dat. Z ručního prohlížení dat byly vyvozeny důsledky, které tvořily základ pro splnění bodu  
1026 čtvrtého. Tomuto a následnému bodu zadání je věnována zbývající část páté kapitoly, která  
1027 je doplněna o přílohy nacházející se na konci tohoto dokumentu. Pro automatické zkoumání  
1028 dat byl využit nástroj RapidMiner, který je společně s dalšími vybranými programy popsán  
1029 na konci kapitoly třetí. Její obsah je zaměřen na popis procesu dolování z dat a na jeho za-  
1030 členění do celého bloku získávání znalostí z databází. Poslednímu bodu zadání je vyhrazena  
1031 část v kapitole čtvrté, která se zabývá případným pokračováním této práce.

1032 Jak je uvedeno výše veškeré body zadání byly splněny. Jako rozšíření zadání bylo  
1033 zpřístupnění možnosti využít jak robota tak i data uložená v databázi pro prezentování  
1034 vlastního statusu na webových stránkách. Příklad tohoto použití je přiložený na kompakt-  
1035 ním disku a také je k dispozici k nahlédnutí na internetové stránce [http://pcpanel-1205.  
1036 fit.vutbr.cz/jabinfo/](http://pcpanel-1205.fit.vutbr.cz/jabinfo/).

1037 Pro nedostatečný počet uživatelů, kteří doposud využívali tohoto robota, a zároveň by  
1038 měli správně nastavené „rozesílání“ rozšířených statusů vedlo k následujícímu rozhodnutí.  
1039 Větší důraz byl kladen na vývoj samotného robota a sbírání i ukládání dat do databáze,  
1040 než na následný proces dolování z dat. Za období čtyř měsíců bylo celkem nasbíráno kolem  
1041 200 MB dat a přes 100 tisíc záznamů.

1042

# 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] Bramer, M.: *Principles of Data mining*. London: Springer, první vydání, 2007, 343 s., ISBN 18-462-8765-0.
- [4] 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.
- [5] Han, J.; Kamber, M.: *Data mining : concepts and techniques*. San Francisco: Morgan Kaufmann Publisher, druhé vydání, 2006, 770 s., ISBN 15-586-0901-6.
- [6] Hildebrand, J.; Saint-Andre, P.: XEP-0080: User Location. [online], 15-09-2009, [cit. 16. května 2011].  
URL <http://xmpp.org/extensions/xep-0080.html>
- [7] Hildebrand, J.; Saint-Andre, P.; Tronçon, R.; aj.: XEP-0115: Entity Capabilities. [online], 26-02-2008, [cit. 16. května 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. 16. května 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., ISBN 80-716-9860-1.
- [11] Meijer, R.; Saint-Andre, P.: XEP-0108: User Activity. [online], 29-10-2008, [cit. 16. května 2011].  
URL <http://xmpp.org/extensions/xep-0108.html>
- [12] Millard, P.; Saint-Andre, P.; Meijer, R.: XEP-0060: Publish-Subscribe. [online], 12-07-2010, [cit. 16. května 2011].  
URL <http://xmpp.org/extensions/xep-0060.html>

- 1073 [13] Mizzi, S.; Saint-Andre, P.: XEP-0292: vCard4 Over XMPP. [online], 02-26-2008, [cit.  
1074 16. května 2011].  
1075 URL <http://xmpp.org/extensions/xep-0292.html>
- 1076 [14] Moore, D.; Wright, W.: *Jabber developer's handbook*. Indianapolis: Sams Publishing,  
1077 první vydání, 2004, 487 s., iISBN 06-723-2536-5.
- 1078 [15] Nemrava, M.; Pospíšil, J.: Dolování dat a jeho aplikace. [online], 2006, [cit. 16. května  
1079 2011].  
1080 URL [http://www.spatial.cs.umn.edu/paper\\_ps/dmchap.pdf](http://www.spatial.cs.umn.edu/paper_ps/dmchap.pdf)
- 1081 [16] Saint-Andre, P.: XEP-0054: vcard-temp. [online], 07-16-2008, [cit. 16. května 2011].  
1082 URL <http://xmpp.org/extensions/xep-0054.html>
- 1083 [17] Saint-Andre, P.: XEP-0092: Software Version. [online], 02-15-2007, [cit. 16. května  
1084 2011].  
1085 URL <http://xmpp.org/extensions/xep-0092.html>
- 1086 [18] Saint-Andre, P.: XEP-0118: User Tune. [online], 30-01-2008, [cit. 16. května 2011].  
1087 URL <http://xmpp.org/extensions/xep-0118.html>
- 1088 [19] Saint-Andre, P.: XEP-0153: vCard-Based Avatars. [online], 16-08-2006, [cit.  
1089 16. května 2011].  
1090 URL <http://xmpp.org/extensions/xep-0153.html>
- 1091 [20] Saint-Andre, P.: XEP-0199: XMPP Ping. [online], 03-06-2009, [cit. 16. května 2011].  
1092 URL <http://xmpp.org/extensions/xep-0199.html>
- 1093 [21] Saint-Andre, P.; Meijer, R.: XEP-0107: User Mood. [online], 29-10-2008, [cit.  
1094 16. května 2011].  
1095 URL <http://xmpp.org/extensions/xep-0107.html>
- 1096 [22] Saint-Andre, P.; Smith, K.: XEP-0163: Personal Eventing Protocol. [online],  
1097 12-07-2010, [cit. 16. května 2011].  
1098 URL <http://xmpp.org/extensions/xep-0163.html>
- 1099 [23] Saint-André, P.: Extensible Messaging and Presence Protocol (XMPP): Core.  
1100 [online], 10-2004, [cit. 16. května 2011].  
1101 URL <http://tools.ietf.org/html/rfc3920>
- 1102 [24] Saint-André, P.: Extensible Messaging and Presence Protocol (XMPP): Instant  
1103 Messaging and Presence. [online], 10-2004, [cit. 16. května 2011].  
1104 URL <http://tools.ietf.org/html/rfc3921>
- 1105 [25] Saint-André, P.; Smith, K.; Troncon, R.: *XMPP : the definitive guide : building  
1106 real-time applications with jabber technologies*. Sebastopol: O'Reilly, první vydání,  
1107 2009, 287 s., iISBN 978-059-6521-264.
- 1108 [26] WWW Stránky: Database Systems. [online], 2007, [cit. 16. května 2011].  
1109 URL  
1110 [http://www.cs.uct.ac.za/mit\\_notes\\_devel/Database/Lates%t/index.html](http://www.cs.uct.ac.za/mit_notes_devel/Database/Lates%t/index.html)

- 1111 [27] WWW Stránky: Last.fm. [online], 02-11-2009 [cit. 16. května 2011].  
1112 URL <http://www.last.fm/>
- 1113 [28] WWW Stránky: RapidMiner. [online], 2011, [cit. 16. května 2011].  
1114 URL <http://rapid-i.com/content/view/181/190/>
- 1115 [29] Řezánková, H.; Húsek, D.; Snášel, V.: *Shluková analýza dat*. Praha: Professional  
1116 Publishing, druhé vydání, 2009, 218 s., iISBN 978-808-6946-818.

## 1117 Příloha A

## 1118 Obsah CD

|                                   |                                   |
|-----------------------------------|-----------------------------------|
| <code>./doc/</code>               | - programová dokumentace          |
| <code>./src/jabinfo</code>        | - zdrojové soubory robota         |
| <code>./src/transformation</code> | - zdrojové soubory transformace   |
| <code>./src/datamining</code>     | - zdrojové soubory data miningu   |
| <code>./tex/</code>               | - zdrojová data technické zprávy  |
| <code>./www/</code>               | - zdrojová data webové prezentace |
| <code>./bp-xsendl00.pdf</code>    | - technická zpráva                |
| <code>./MANUAL</code>             | - manuál k programu jabinfo       |
| <code>./README</code>             | - základní informace              |

## 1119 Příloha B

# 1120 Slovník zkratek

1121 **Base64** — převedení binárních dat pomocí znaků ASCII.

1122 **CSV** (Comma-separated values) — souborový formát určený pro výměnu dat z tabulek.

1123 **GPG** (GNU Privacy Guard) — slouží k podepisování a kontrolu podpisu různých dat.

1124 **ID3v1** — datový formát obsahující informace o skladbě.

1125 **IM služby** (Instant messaging) — umožňují posílat zprávy a komunikovat mezi uživateli.

1126 **Jabber** — viz XMPP.

1127 **JEP** (Jabber Enhancement Proposal) — starší název pro XEP.

1128 **JID** (Jabber ID) — jednoznačný identifikátor v síti Jabber.

1129 **SASL** (Simple Authentication and Security Layer) — metoda sloužící ověřování klientů  
1130 na serverech.

1131 **Stanza** — XML stream.

1132 **SQL** (Structured Query Language) — jazyk používaný pro komunikaci s databázemi.

1133 **TLS** (Transport Layer Security) — kryptografický protokol, poskytuje zabezpečenou ko-  
1134 munikaci na internetu.

1135 **TSQL** (Transact-SQL) — jazyk pro temporální databáze.

1136 **vCard** — elektronická osobní vizitka.

1137 **XEP** (XMPP Extension Protocol) — rozšiřuje protokol RFC.

1138 **XML** (Extensible Markup Language) — univerzální značkový jazyk.

1139 **XMPP** (Extensible Messaging and Presence Protocol) — protokol pro doručování zpráv.



## 1140 Příloha C

### 1141 Stanza - základní schéma

1142 Přehled základních elementů, které jsou využívány při Jabber komunikaci. Struktura jed-  
1143 notlivých částí stanzy ukazuje pouze prvky relativní k této práci. Pomocí hranatých závorek  
1144 je znázorněna množina, ze které musí být vybrán právě jeden prvek. Na místě uvozovek se  
1145 očekává jakákoliv povolená hodnota.

#### 1146 Iq

---

```
1      <iq from=""  
2          to=""  
3          type="[ get , set , result , error ]"  
4          id=""  
5          Namespace  
6      </iq>
```

---

Algoritmus C.1: Popis elementu *iq*.

#### 1147 Message

---

```
1      <message from=""  
2          to=""  
3          type="[ normal , chat , groupchat , headline , error ]"  
4          id=""  
5          <body> </body>  
6          <x xmlns="jabber:x:event">  
7              [ Offline , Delivered , Displayed , Composing ]  
8          <subject> </subject>  
9          <thread> </thread>  
10         <error> </error>  
11         <x> </x>  
12 </message>
```

---

Algoritmus C.2: Popis elementu *message*.

---

```

1  <presence from=""
2      to=""
3      type="[available , unavailable , probe , subscribe ,
4            unsubscribe , subscribed , unsubscribed , error]"
5      id=""
6  <show>
7      [away , chat , dnd , normal , xa]
8  </show>
9  <status>    </status>
10 <priority>   </priority>
11 <error>      </error>
12 </presence>

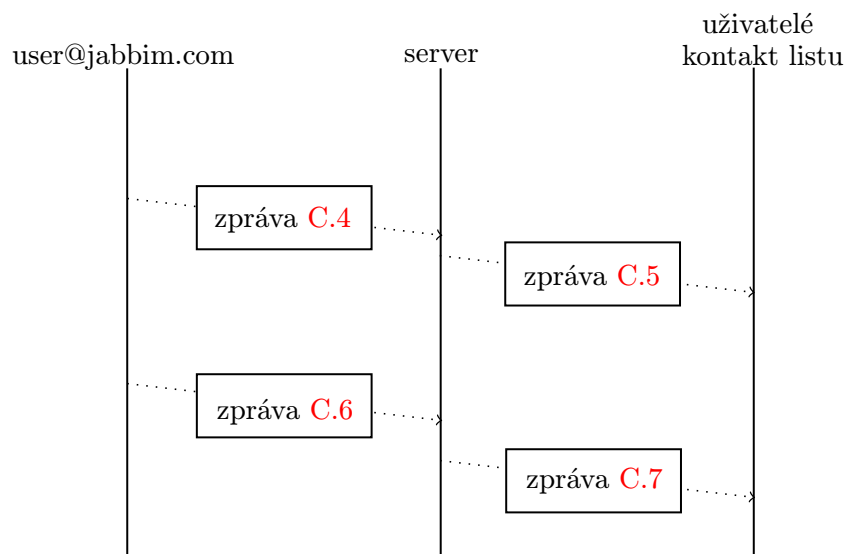
```

---

Algoritmus C.3: Popis elementu *presence*.

## 1149 Přehled průběhu rozšíření

1150 Ukázka celého příkladu šíření statusu pomocí rozšíření *User Tune*. Uživatel *user* poslouchá hudbu a informuje server zasláním zprávy zobrazené v příkladu C.4.

Obrázek C.1: Ukázka „šíření“ *User Tune*.

---

```

1  <iq from="user@jabber.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>

```

---

Algoritmus C.4: Informování serveru o právě přehrávané hudbě.

1152 Server obdrží informace od klienta *user* prostřednictvím zprávy, jejíž obsah je napl-  
1153 něn daty spojenými s přehrávanou hudbou. Pomocí elementu *message* ji přepośle všem  
1154 uživatelům ze seznamu kontaktů uživatele *user*, kteří se pro odběr těchto typů zpráv zare-  
gistrovali. Struktura této zprávy je prezentována na příkladu C.5.

---

```

1  <message from="user@jabber.com" type="set"
2    to="jabinfo@jabber.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            <source>Nigredo</source>
10           <title>1968</title>
11           <track>5</track>
12         </tune>
13       </item>
14     </items>
15   </event>
16 </message>

```

---

Algoritmus C.5: Server informuje uživatele podporující rozšíření o stavu *user@jabber.com*.

1155

1156 Zpráva o přehrávané hudbě je také přeposlána všem otevřeným spojením uživatele *user*,  
ukázáno na příkladu C.6.

---

```
1 <message from="user@jabbbim.com" type="set"
2 to="user@jabbbim.com/дома" id="pub2">
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 <source>Nigredo</source>
10 <title>1968</title>
11 <track>5</track>
12 </tune>
13 </item>
14 </items>
15 </event>
16 </message>
```

---

Algoritmus C.6: Server přepoše informace o přehrávané hudbě všem otevřeným spojením uživatele *user@jabbbim.com*.

1157  
1158 Přestane-li uživatel *user* poslouchat/vysílat informace o přehrávané hudbě, provede to  
1159 pomocí zprávy ukázané na příkladu C.7. Zpráva typu *iq*, ve které je položka *tune*, nesoucí  
1160 informace o skladbě, je prázdná.

---

```
1161 1 <iq from="user@jabbbim.com/prace" type="set" id="pub1">
1162 2 <pubsub xmlns="http://jabber.org/protocol/pubsub">
1163 3 <publish node="http://jabber.org/protocol/tune">
1164 4 <item>
1165 5 <tune xmlns="http://jabber.org/protocol/tune"/>
1166 6 </item>
1167 7 </publish>
1168 8 </pubsub>
1169 9 </iq>
1170
```

---

Algoritmus C.7: Uživatel ukončil „vysílání“ rozšířených zpráv o svém stavu.

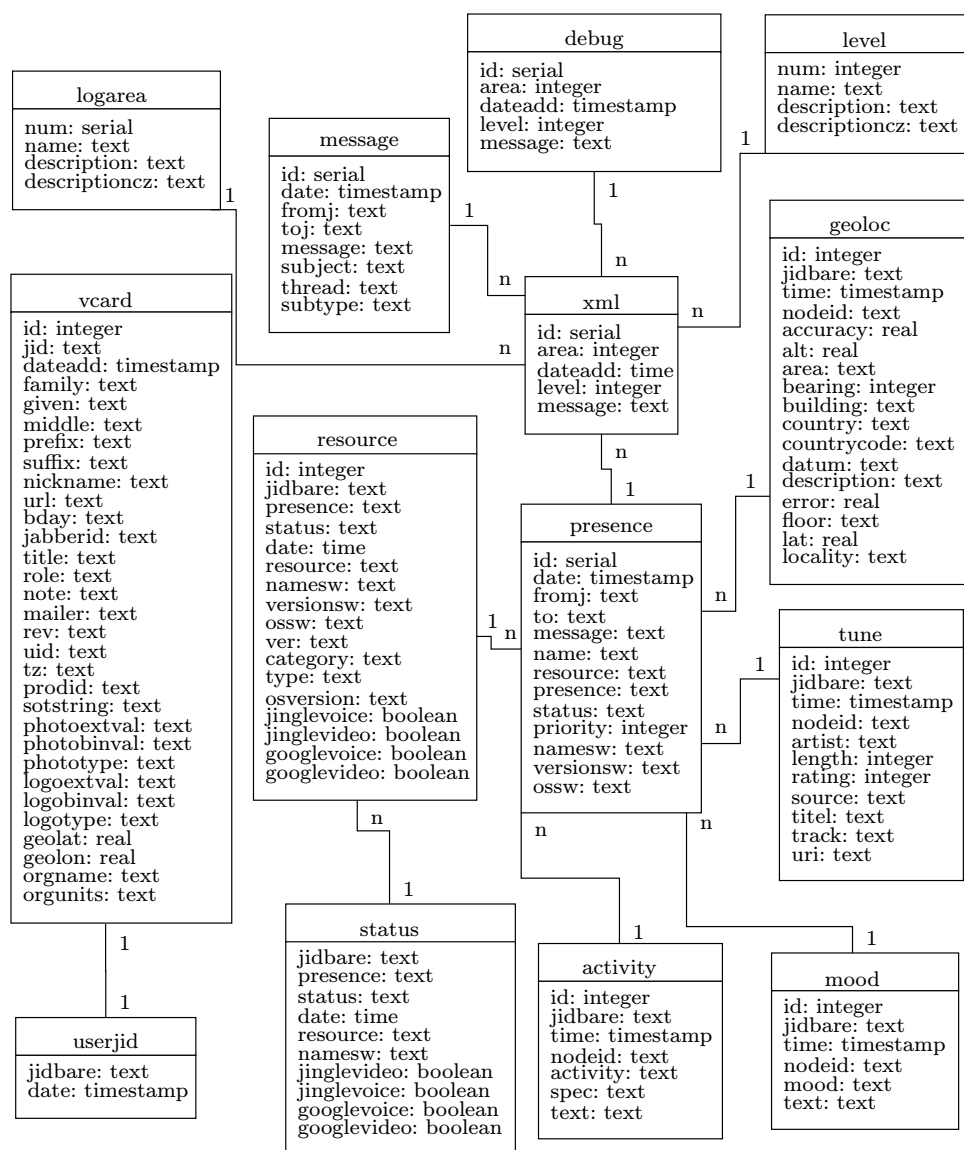
1172 Server informuje všechny účastníky odběru zprávou, která má položku *tune* prázdnou.  
1173 Tak jak to prezentuje příklad C.8.

---

```
1174 1 <message from="user@jabbbim.com"
1175 2 to="jabinfo@jabbbim.com/bot">
1176 3 <event xmlns="http://jabber.org/protocol/pubsub#event">
1177 4 <items node="http://jabber.org/protocol/tune">
1178 5 <item>
1179 6 <tune xmlns="http://jabber.org/protocol/tune"/>
1180 7 </item>
1181 8 </items>
1182 9 </event>
1183 10 </message>
1184
```

---

Algoritmus C.8: Server informuje klienty o ukončení šíření rozšířeného statusu.



Obrázek D.1: Struktura databáze

## 1188 Příloha E

### 1189 Přehled příkazů robota

1190 Přehled základních zpráv obsahující příkazy pro robota a jejich vysvětlení je prezentováno  
1191 v tabulce E.1. Kde první sloupec značí příkaz, který rozlišuje velká a malá písmena, a druhý  
1192 jej krátce a výstižně popisuje. V případě, že uživatel nemá žádný údaj v databázi potřebný  
1193 k vyhodnocení dotazu, je zaslána pouze informativní zpráva, které vzniklý problém vy-  
1194 světluje. V případě zaslání neznámého dotazu, je uživateli automaticky odeslána zpráva  
1195 s obsahem příkazu HELP.

| příkaz   | vysvětlení                        |
|----------|-----------------------------------|
| HELP     | Přehled příkazů a jejich popis.   |
| INFO     | Popis aplikace.                   |
| HISTORY  | Výpis posledních 10 změn statusů. |
| TUNE     | Výpis o přehrávané hudbě.         |
| GEOLOC   | Výpis o geografické poloze.       |
| MOOD     | Výpis o aktuální náladě.          |
| ACTIVITY | Výpis o aktuální činnosti.        |
| VCARD    | Základní obsah VCard              |

Tabulka E.1: Přehled příkazů pro robota.

## 1196 Příloha F

### 1197 Přehled vztahů uživatelů

1198 Tabulkou [F.1](#) je prezentován vztah mezi uživatelským JID a náhodně přiřazeným číslem,  
1199 které jej v této práci zastupuje. Klientské jednoznačné identifikační názvy jsou zde zobra-  
1200 zeny bez části, která následuje za znakem zavináč, tedy bez názvu serveru. K tomuto kroku  
1201 bylo přistoupeno ke snaze zajistit alespoň minimální stupeň anonymity.

| JID            | číslo | JID         | číslo |
|----------------|-------|-------------|-------|
| portilo        | 1     | mewoack     | 22    |
| xsendl00       | 2     | yac         | 23    |
| hrabos         | 3     | komi        | 24    |
| kubira         | 4     | drake127    | 25    |
| mautinek       | 5     | fletcher    | 26    |
| danulka        | 6     | boky        | 27    |
| matmatmat      | 7     | mtmccox     | 28    |
| ybyzak         | 8     | tux.martin  | 29    |
| vecerniceverca | 9     | martin      | 30    |
| pojir          | 10    | tux.martin  | 31    |
| vtheman        | 11    | xapoh       | 32    |
| jurij.h        | 12    | matasx      | 33    |
| imlich         | 13    | javier.brat | 34    |
| joejoe         | 14    | sychra      | 35    |
| semal          | 15    | ondrg       | 36    |
| mr.aston       | 16    | moen        | 37    |
| mrazek.v       | 17    | myler       | 38    |
| theslayer      | 18    | martix      | 39    |
| kuba86         | 19    | rezza       | 40    |
| minimax        | 20    | ales.lanik  | 41    |
| miklik         | 21    |             |       |

Tabulka F.1: Zobrazení uživatelského JID na čísla.

1202 Tabulka [F.2](#), která se nachází níže, plně zobrazuje procentuální shodu mezi jednotlivými  
1203 uživateli. Čísla v prvním řádku a sloupci odpovídají přiřazení prezentovanému v tabulce [F.1](#).  
1204 Barevně odlišené buňky tabulky vyzdvihují větší procentuální shodu než jaká je u ostatních  
1205 hodnot. Barvě zelené odpovídá interval od 80 % – 89 % a pro barvu červenou 90 % – 99 %.

1206

|    |    |           |    |    |           |           |           |    |    |    |    |           |    |    |    |    |    |           |           |    |    |    |    |      |    |    |    |    |           |           |           |    |    |    |    |           |           |           |           |           |           |
|----|----|-----------|----|----|-----------|-----------|-----------|----|----|----|----|-----------|----|----|----|----|----|-----------|-----------|----|----|----|----|------|----|----|----|----|-----------|-----------|-----------|----|----|----|----|-----------|-----------|-----------|-----------|-----------|-----------|
| -  | 1  | 2         | 3  | 4  | 5         | 6         | 7         | 8  | 9  | 10 | 11 | 12        | 13 | 14 | 15 | 16 | 17 | 18        | 19        | 20 | 21 | 22 | 23 | 24   | 25 | 26 | 27 | 28 | 29        | 30        | 31        | 32 | 33 | 34 | 35 | 36        | 37        | 38        | 39        | 40        | 41        |
| 1  | -  | 22        | 31 | 70 | 23        | 28        | 31        | 37 | 55 | 54 | 64 | 14        | 52 | 32 | 34 | 54 | 44 | 27        | 23        | 19 | 29 | 23 | 60 | 52   | 51 | 25 | 29 | 53 | 33        | 34        | 35        | 61 | 49 | 40 | 53 | 32        | 23        | 27        | 14        | 26        | 22        |
| 2  | 22 | -         | 63 | 24 | <b>90</b> | <b>88</b> | <b>84</b> | 55 | 52 | 39 | 34 | <b>90</b> | 29 | 14 | 56 | 32 | 44 | <b>89</b> | <b>85</b> | 50 | 33 | 22 | 31 | 36   | 43 | 36 | 29 | 34 | 53        | 61        | 60        | 33 | 29 | 25 | 49 | <b>81</b> | <b>84</b> | 48        | <b>91</b> | 77        | <b>92</b> |
| 3  | 31 | 63        | -  | 26 | 63        | 66        | 64        | 57 | 55 | 47 | 42 | 60        | 45 | 22 | 79 | 34 | 44 | 66        | 63        | 57 | 36 | 38 | 36 | 45   | 58 | 53 | 51 | 37 | 51        | 58        | 58        | 37 | 36 | 29 | 55 | 69        | 65        | 57        | 60        | 68        | 63        |
| 4  | 70 | 24        | 26 | -  | 23        | 29        | 30        | 36 | 53 | 48 | 58 | 14        | 42 | 26 | 28 | 58 | 41 | 22        | 18        | 30 | 19 | 45 | 42 | 42   | 21 | 53 | 31 | 33 | 36        | 55        | 42        | 36 | 47 | 29 | 23 | 15        | 25        | 21        |           |           |           |
| 5  | 23 | <b>90</b> | 63 | 23 | -         | <b>90</b> | <b>85</b> | 60 | 49 | 42 | 34 | <b>88</b> | 32 | 17 | 55 | 30 | 42 | <b>86</b> | <b>86</b> | 49 | 34 | 23 | 29 | 36   | 41 | 34 | 31 | 35 | 53        | 58        | 58        | 34 | 29 | 27 | 48 | <b>75</b> | <b>82</b> | 49        | <b>88</b> | 79        | <b>91</b> |
| 6  | 28 | <b>88</b> | 66 | 29 | <b>90</b> | -         | <b>86</b> | 60 | 56 | 44 | 38 | <b>85</b> | 34 | 17 | 58 | 36 | 42 | <b>87</b> | <b>85</b> | 49 | 32 | 24 | 34 | 37   | 42 | 34 | 31 | 34 | 51        | 57        | 56        | 38 | 34 | 31 | 51 | <b>77</b> | <b>84</b> | 49        | <b>85</b> | <b>90</b> | <b>86</b> |
| 7  | 31 | <b>84</b> | 64 | 30 | <b>85</b> | <b>86</b> | -         | 59 | 53 | 49 | 43 | <b>81</b> | 31 | 22 | 56 | 32 | 42 | <b>86</b> | <b>83</b> | 51 | 34 | 25 | 33 | 44   | 49 | 36 | 32 | 39 | 53        | 62        | 61        | 41 | 37 | 36 | 54 | 79        | <b>82</b> | 51        | <b>81</b> | <b>80</b> | <b>83</b> |
| 8  | 37 | 55        | 57 | 36 | 60        | 60        | 59        | -  | 60 | 53 | 48 | 51        | 41 | 21 | 60 | 39 | 51 | 58        | 59        | 50 | 43 | 24 | 41 | 42   | 51 | 34 | 33 | 42 | 60        | 66        | 66        | 50 | 42 | 31 | 55 | 60        | 55        | 56        | 51        | 58        | 54        |
| 9  | 55 | 52        | 55 | 53 | 49        | 56        | 53        | 60 | -  | 58 | 58 | 43        | 53 | 26 | 55 | 52 | 63 | 56        | 52        | 47 | 43 | 26 | 60 | 51   | 60 | 40 | 37 | 56 | 62        | 62        | 51        | 43 | 36 | 65 | 56 | 51        | 52        | 44        | 53        | 49        |           |
| 10 | 54 | 39        | 47 | 48 | 42        | 44        | 49        | 53 | 58 | -  | 60 | 33        | 49 | 31 | 48 | 55 | 56 | 44        | 42        | 39 | 44 | 29 | 58 | 61   | 62 | 42 | 45 | 61 | 44        | 46        | 47        | 59 | 54 | 47 | 66 | 50        | 45        | 34        | 48        | 41        |           |
| 11 | 64 | 34        | 42 | 58 | 34        | 38        | 43        | 48 | 58 | 60 | -  | 26        | 48 | 35 | 44 | 55 | 46 | 38        | 34        | 31 | 44 | 29 | 58 | 61   | 60 | 36 | 41 | 59 | 41        | 44        | 44        | 64 | 55 | 48 | 58 | 44        | 32        | 38        | 27        | 39        | 34        |
| 12 | 14 | <b>90</b> | 60 | 14 | <b>88</b> | <b>85</b> | <b>81</b> | 51 | 43 | 33 | 26 | -         | 24 | 9  | 51 | 25 | 36 | <b>82</b> | <b>82</b> | 49 | 29 | 21 | 24 | 29   | 34 | 32 | 27 | 26 | 47        | 54        | 53        | 27 | 24 | 23 | 42 | 73        | <b>81</b> | 46        | <b>99</b> | 75        | <b>88</b> |
| 13 | 52 | 29        | 45 | 42 | 32        | 34        | 31        | 41 | 53 | 49 | 48 | 24        | -  | 29 | 45 | 56 | 53 | 37        | 29        | 29 | 40 | 30 | 66 | 50   | 50 | 34 | 38 | 58 | 34        | 37        | 38        | 43 | 43 | 36 | 49 | 38        | 31        | 24        | 42        | 32        |           |
| 14 | 32 | 14        | 22 | 26 | 17        | 17        | 22        | 21 | 26 | 31 | 35 | 9         | 29 | -  | 24 | 25 | 22 | 17        | 14        | 16 | 23 | 19 | 26 | 42   | 32 | 19 | 25 | 27 | 15        | 16        | 16        | 24 | 25 | 26 | 25 | 20        | 16        | 10        | 21        | 15        |           |
| 15 | 34 | 56        | 79 | 28 | 55        | 58        | 56        | 60 | 55 | 48 | 44 | 51        | 45 | 24 | -  | 37 | 49 | 59        | 56        | 59 | 39 | 44 | 38 | 50   | 55 | 59 | 38 | 51 | 57        | 37        | 34        | 42 | 34 | 56 | 62 | 56        | 60        | 51        | 58        | 56        |           |
| 16 | 54 | 32        | 34 | 58 | 30        | 36        | 32        | 39 | 52 | 55 | 55 | 25        | 56 | 25 | 37 | -  | 49 | 34        | 30        | 27 | 36 | 25 | 58 | 47   | 48 | 32 | 33 | 58 | 33        | 38        | 53        | 48 | 38 | 56 | 41 | 34        | 29        | 25        | 38        | 32        |           |
| 17 | 44 | 44        | 44 | 41 | 42        | 42        | 42        | 51 | 63 | 56 | 46 | 36        | 53 | 22 | 49 | 49 | -  | 49        | 46        | 38 | 49 | 25 | 49 | 51   | 60 | 36 | 34 | 58 | 59        | 58        | 58        | 40 | 34 | 28 | 54 | 52        | 46        | 44        | 36        | 49        | 42        |
| 18 | 27 | <b>89</b> | 66 | 26 | <b>86</b> | <b>87</b> | <b>86</b> | 58 | 56 | 44 | 38 | <b>82</b> | 37 | 17 | 59 | 34 | 49 | -         | <b>86</b> | 51 | 36 | 23 | 36 | 43   | 49 | 36 | 31 | 38 | 55        | 64        | 64        | 36 | 31 | 29 | 52 | <b>82</b> | <b>84</b> | 49        | <b>83</b> | <b>81</b> | <b>88</b> |
| 19 | 23 | <b>85</b> | 63 | 21 | <b>86</b> | <b>85</b> | <b>83</b> | 59 | 52 | 42 | 34 | <b>82</b> | 29 | 14 | 56 | 30 | 46 | <b>86</b> | -         | 51 | 36 | 25 | 34 | 36   | 45 | 36 | 33 | 34 | 57        | 65        | 65        | 36 | 32 | 27 | 45 | <b>77</b> | <b>83</b> | <b>86</b> | <b>82</b> | <b>77</b> | <b>86</b> |
| 20 | 19 | 50        | 57 | 18 | 49        | 49        | 51        | 50 | 47 | 39 | 31 | 49        | 29 | 16 | 59 | 27 | 38 | 51        | 51        | -  | 34 | 35 | 27 | 40   | 44 | 52 | 42 | 27 | 49        | 51        | 51        | 30 | 36 | 36 | 44 | 52        | 51        | 51        | 49        | 51        | 52        |
| 21 | 29 | 33        | 36 | 30 | 34        | 32        | 34        | 43 | 43 | 44 | 44 | 29        | 40 | 23 | 39 | 36 | 49 | 36        | 36        | 34 | -  | 26 | 37 | 45   | 51 | 36 | 36 | 42 | 47        | 43        | 43        | 36 | 31 | 33 | 37 | 36        | 36        | 29        | 35        | 36        |           |
| 22 | 23 | 22        | 38 | 19 | 23        | 24        | 25        | 24 | 26 | 29 | 21 | 30        | 19 | 44 | 25 | 25 | 23 | 25        | 35        | 26 | -  | 26 | 30 | 31   | 70 | 57 | 25 | 23 | 23        | 23        | 25        | 44 | 36 | 27 | 26 | 24        | 35        | 21        | 26        | 25        |           |
| 23 | 60 | 31        | 36 | 45 | 29        | 34        | 33        | 41 | 60 | 58 | 58 | 24        | 66 | 26 | 38 | 58 | 49 | 36        | 34        | 27 | 37 | 26 | -  | 56   | 55 | 31 | 36 | 56 | 34        | 38        | 40        | 59 | 52 | 42 | 56 | 40        | 37        | 35        | 25        | 40        | 32        |
| 24 | 52 | 36        | 45 | 42 | 36        | 37        | 44        | 42 | 51 | 61 | 61 | 29        | 50 | 42 | 50 | 47 | 51 | 43        | 36        | 40 | 45 | 30 | 56 | -    | 69 | 45 | 44 | 49 | 43        | 47        | 48        | 51 | 47 | 47 | 55 | 47        | 34        | 37        | 30        | 44        | 36        |
| 25 | 51 | 43        | 48 | 42 | 41        | 42        | 49        | 51 | 60 | 62 | 60 | 34        | 50 | 32 | 55 | 48 | 60 | 49        | 45        | 44 | 51 | 31 | 55 | 69   | -  | 48 | 44 | 51 | 53        | 58        | 59        | 55 | 48 | 46 | 55 | 53        | 42        | 48        | 34        | 44        | 44        |
| 26 | 25 | 36        | 53 | 21 | 34        | 34        | 36        | 34 | 40 | 42 | 36 | 32        | 34 | 19 | 59 | 32 | 36 | 36        | 36        | 52 | 36 | 70 | 31 | 45   | 48 | -  | 66 | 31 | 37        | 38        | 38        | 34 | 50 | 40 | 39 | 39        | 35        | 45        | 33        | 36        | 38        |
| 27 | 29 | 29        | 51 | 23 | 31        | 31        | 32        | 33 | 37 | 45 | 41 | 27        | 38 | 25 | 59 | 33 | 34 | 31        | 33        | 42 | 36 | 57 | 36 | 44   | 44 | 66 | -  | 38 | 31        | 31        | 31        | 36 | 51 | 42 | 38 | 36        | 34        | 42        | 27        | 33        | 31        |
| 28 | 53 | 34        | 37 | 51 | 35        | 34        | 39        | 42 | 56 | 61 | 59 | 26        | 58 | 27 | 38 | 58 | 38 | 34        | 27        | 42 | 25 | 56 | 49 | 51   | 31 | 38 | -  | 43 | 40        | 40        | 54        | 49 | 43 | 59 | 40 | 43        | 35        | 27        | 42        | 33        |           |
| 29 | 33 | 53        | 51 | 33 | 53        | 51        | 53        | 60 | 62 | 44 | 41 | 47        | 34 | 15 | 51 | 33 | 59 | 55        | 57        | 49 | 47 | 23 | 34 | 43   | 53 | 37 | 31 | 43 | -         | <b>80</b> | 79        | 43 | 31 | 25 | 49 | 55        | 48        | 55        | 47        | 49        | 50        |
| 30 | 34 | 61        | 58 | 36 | 58        | 57        | 62        | 66 | 62 | 46 | 44 | 54        | 37 | 16 | 57 | 38 | 58 | 64        | 65        | 51 | 43 | 23 | 38 | 47   | 58 | 38 | 31 | 40 | <b>80</b> | -         | <b>98</b> | 45 | 36 | 29 | 52 | 65        | 56        | 54        | 55        | 58        | 59        |
| 31 | 35 | 60        | 58 | 36 | 58        | 56        | 61        | 66 | 62 | 47 | 44 | 53        | 38 | 16 | 57 | 38 | 58 | 64        | 65        | 51 | 43 | 23 | 40 | 48   | 59 | 38 | 31 | 40 | 79        | <b>98</b> | -         | 45 | 36 | 30 | 52 | 65        | 55        | 54        | 54        | 58        | 58        |
| 32 | 61 | 33        | 37 | 55 | 34        | 38        | 41        | 50 | 51 | 59 | 64 | 27        | 43 | 24 | 34 | 53 | 40 | 36        | 36        | 30 | 36 | 25 | 59 | 51   | 55 | 34 | 36 | 54 | 43        | 45        | 45        | -  | 65 | 51 | 63 | 41        | 32        | 37        | 27        | 35        | 33        |
| 33 | 49 | 29        | 36 | 42 | 29        | 34        | 37        | 42 | 43 | 54 | 55 | 24        | 43 | 25 | 42 | 48 | 34 | 31        | 32        | 36 | 31 | 44 | 52 | 47   | 48 | 50 | 51 | 49 | 31        | 36        | 65        | -  | 56 | 53 | 36 | 29        | 40        | 25        | 31        | 31        |           |
| 34 | 40 | 25        | 29 | 36 | 27        | 31        | 36        | 31 | 36 | 47 | 48 | 23        | 36 | 26 | 34 | 38 | 28 | 29        | 27        | 36 | 31 | 36 | 42 | 47   | 46 | 40 | 42 | 43 | 25        | 29        | 30        | 51 | 56 | -  | 49 | 29        | 27        | 33        | 23        | 29        | 29        |
| 35 | 53 | 49        | 55 | 47 | 48        | 51        | 54        | 55 | 65 | 66 | 58 | 42        | 49 | 25 | 56 | 56 | 54 | 52        | 45        | 44 | 33 | 27 | 56 | 55   | 55 | 39 | 38 | 59 | 49        | 52        | 52        | 63 | 53 | 49 | -  | 58        | 49        | 51        | 43        | 58        | 47        |
| 36 | 32 | <b>81</b> | 69 | 29 | 75        | 77        | 79        | 60 | 56 | 50 | 44 | 73        | 38 | 20 | 62 | 41 | 52 | <b>82</b> | <b>87</b> | 52 | 37 | 26 | 40 | 47   | 53 | 39 | 36 | 40 | 55        | 65        | 41        | 36 | 29 | 58 | -  | 77        | 51        | 73        | <b>81</b> | 78        |           |
| 37 | 23 | <b>84</b> | 65 | 23 | <b>82</b> | <b>84</b> | <b>82</b> | 55 | 51 | 45 | 32 | <b>81</b> | 38 | 16 | 56 | 34 | 46 | <b>84</b> | <b>83</b> | 51 | 36 | 24 | 37 | 34   | 42 | 35 | 34 | 43 | 48        | 56        | 55        | 32 | 29 | 27 | 49 | 77        | -         | 47        | <b>81</b> | <b>82</b> | <b>84</b> |
| 38 | 27 | 48        | 57 | 23 | 49        | 49        | 51        | 56 | 52 | 45 | 38 | 46        | 31 | 16 | 60 | 29 | 44 | 49        | 56        | 51 | 36 | 35 | 37 | 48   | 45 | 42 | 35 | 55 | 54        | 54        | 37        | 40 | 33 | 51 | 51 | 47        | -         | 46        | 49        | 49        |           |
| 39 | 14 | <b>91</b> | 60 | 15 | <b>88</b> | <b>85</b> | <b>81</b> | 51 | 44 | 34 | 27 | <b>99</b> | 24 | 10 | 51 | 25 | 36 | <b>83</b> | <b>82</b> | 49 | 29 | 21 | 25 | 30   | 34 | 33 | 27 | 27 | 47        | 55        | 54        | 27 | 25 | 23 | 43 | 73        | <b>81</b> | 46        | -         | 75        | <b>89</b> |
| 40 | 26 | 77        | 68 | 25 | 79        | <b>80</b> | <b>80</b> | 58 | 53 | 48 | 39 | 75        | 42 | 21 | 58 | 38 | 49 | <b>81</b> | 77        | 51 | 35 | 26 | 44 | 44</ |    |    |    |    |           |           |           |    |    |    |    |           |           |           |           |           |           |