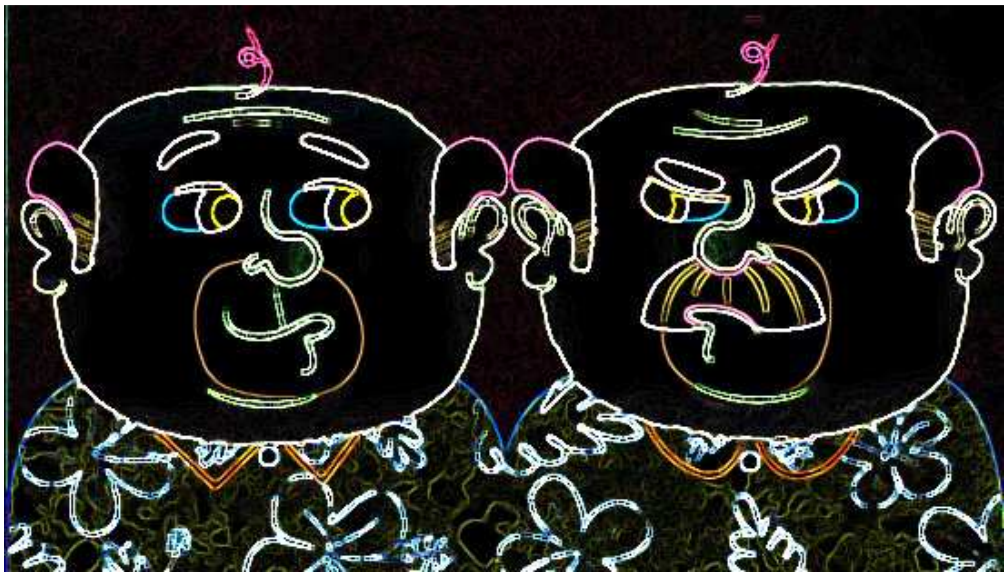
Evaluation of Similarity Metrics

Supporting matching of patients with cardiovascular disease



Author: **Gintas Palionis**

Date of Colloquium: **January 28th, 2011**

Graduation committee:

dr. ir. B.J.F. van Beijnum: RMT/BSS/EWI

dr. M.E. Iacob: IS&CM/MB

Summary

The purpose of this thesis was to support the matching process among user profiles in health care virtual communities. We elaborated in three different aspects. First, we designed and applied a methodology to evaluate software packages and similarity metrics. Second, we designed a patient information model for CVD patients. This model was used in the *Patient Comparison System*. Third, we investigated the weights for properties in the patient information model.

We developed the *Patient Comparison System* to be able to apply the designed patient information model, implement semantic-based similarity metrics, and to help to analyze the weights for properties in the patient information model. The main objective of PCS was to calculate similarity among user profiles.

Based on our results we can draw few important conclusions. First, we concluded that the software package "SimMetrics" shows better performance over the package "SecondString". We compared the source code and the calculation results between similarity metrics that are implemented in both packages and have equal names. Second, two syntactic similarity metrics *Mean Length* and *Levenstein* outperformed other during the experiments. We used people opinion to decide which metric performs better. Third, we saw that the Patient Comparison System calculated meaningful matches among users. The calculated similarities by the system and the assigned similarities by users have the correlation in the range [0.349; 0.511].

Table of Contents

Table of Figures

Table of Tables

1 Introduction

This chapter presents the motivation, the research problem, objectives, research questions, and research approach of this thesis.

1.1 Motivation

The Internet provides people with a new medium for social activities thereby making possible entirely new features of social reality (Kim, Leeb and Hiemstra, 2003). The number of Internet users is expected to reach almost two billion in 2010 (Internet Usage Statistics). Most of the users are members of some kind of virtual community (VC). To identify themselves, members of virtual communities use user profiles which represent characteristics of users.

VCs are also applied in health care where they help to conduct health-care-related activities like delivery of health care services, education, giving support, sharing problems and interests (Demiris, 2006). Stakeholders of such communities are patients, caregivers, nurses, hospital managers, etc. Furthermore, VCs can be disease-specific. For example, among others, the cardiovascular disease (CVD) is also the subject of virtual communities. CVD is one of the major illnesses which took live of 17.1 million people in 2004 worldwide. It is the number one cause of death in the world.

Past studies show that exercise sessions are beneficial to reduce illness development for CVD patients. Furthermore, exercising in groups shows even more benefits. As the name implies, group exercise sessions bring together people in one place where they can increase muscle endurance and functional mobility (Rubenstein et al., 2000). Mutrie et al. (2007) also argue that group sessions show benefits in terms of physical and psychological functioning. Williams and Lord (1997) add that during group exercise sessions patients experience social interaction, enjoyment and supervision.

In the recent years the number of scientific publications about the benefits of exercise sessions for patient with CVD has increased. However, these publications vary in the way they perform scientific trials and experiments. For example, different studies focus on different types of exercises, number of exercise sessions per week, or length of exercise session. Moreover, these studies aim to present results which show benefits in terms of increased health status of patients. However, there is lack of knowledge about the uniformity of group exercise sessions. With respect to this, there is a need to make a conceptual model which could characterize the group exercise session. Such a model could be used by VC to perform different tasks.

To make VCs more beneficial for users matching services can help. Simply speaking, matching services compare VC users according to the information in user profiles. Matching of people in

VCs can lead to a better knowledge sharing, communication, support giving and social interactions (Terveen and McDonald, 2005).

In the field of computer science, the matching problem is not new; however, the application domain is rapidly emerging. It is widely discussed in semantic web, web-services, peer-to-peer systems, information integration, etc. (Euzenat and Shvailo, 2007). Thus, it is expected that in the future more examples of applications will appear. Matching of people is also the object of interest in the area of information filtering, recommender systems and collaborative filtering.

Shardanand and Maes (1995) describe the idea of profile matching in three steps:

- VC maintains a user profile.
- VC compares this profile to other profiles of other users.
- VC considers the most similar profiles to perform specific tasks.

The first and second steps are the most interesting for us in this thesis. Considering the importance of health care VCs and CVD, we are interested in designing the user profile for patients with CVD. Furthermore, the comparison process of user profiles is the second focus of this study. Various algorithms are used to compare profiles (Brozovsky and Petricek, 2007); however, there is a lack of scientific publications which deal with metrics that are used to calculate similarity among user profiles. Moreover, in the area of online dating, matching services return either only few matches or a huge number of similar profiles (Brozovsky and Petricek, 2007). The problem might be that VC or users assign contradicting weights to properties in the user profile. It is easy to understand that various properties in the user profile might be assigned to different weights, which depend on the matching requirements and context. For example, different users understand the importance of property “hobbies” differently. This might lead to irrelevant matching results for a user. With respect to this, we investigate weights of user profile properties that influence the result of matching among profiles.

To address the issues mentioned above, we start with designing the user information model for patients with CVDs which can be represented by the user profile. During the process of design, we investigate various frameworks that are used in user modeling and basic characteristics of group exercise sessions. These characteristics can be applied to the majority of exercise sessions for patients with CVD. Furthermore, we analyze the problem of finding fellow patients for the same exercise session. Based on past studies, we make an assumption that, patients with most similar preferences for groups exercise sessions should participate in the same session. Moreover, personal characteristics of patients play also an important role in our matching process.

Then we investigate similarity metrics which calculate the similarity between user profiles. Various similarity metrics in data and information retrieval deal with vector-based representation of user profiles. Past studies investigated the similarity metrics and some

resulted in software packages which implemented them. Furthermore, these software packages were used in a number of scientific experiments in other scientific fields such as biology; however, usually, authors do not provide enough motivation for choosing one or another software package. Moreover, there is a lack of knowledge about evaluation of metrics and software packages in terms of user perceived opinion.

For the analysis, we take software implementations of various metrics, and compare them in terms of input types and calculated results. Next, we investigate weights of each property in the user profile. We aim to find the best performing set of weights. Finally, the Patient Comparing System (PCS) is developed to employ designed patient information model, the weights for properties in the user profile, and investigated similarity metrics. This system is also used to evaluate matching results among users who are assumed to play the role of patient with CVDs.

We are now in a position to analyze these challenges and the following sections present deeper research insights and objectives.

1.2 Problem Definition

User modeling or user profiling became very popular in the field of information systems. User profiles represent user competences or preferences as well as other characteristics of the users (Razmerita, 2007). It is also argued that building a good system in which a humans and machine cooperate properly requires taking into account characteristics of people (Rich, 1983). These characteristics can be represented by user model.

The user modeling is either knowledge-based or behavior-based (Middleton, Shadbolt, and De Roure, 2004). Knowledge-based approach builds static model of user. Questionnaires and interviews are often used to obtain the user knowledge. Behavior-based approach uses the user's behavior as a model. Thus, machine-learning techniques are used to discover useful patterns in the user behavior.

Every user profile stores information about the member of VC. This information can be seen as a set of properties and their values assigned to the user. These properties can be of various data types and values can represent any kind of information. Properties of a user can be designed as term-based vectors (Rich, 1983) or semantic structures such as ontologies (Sieg, Mobasher and Burke, 2007). The first approach takes user profile as a set of characteristics assigned to the user. Besides that, the second approach introduces relationships between characteristics, usually, in a representation of a tree. We focus on vector-based approach because we assume that every user (patient) in the VC would have the same user profile. Then the matching between two users would lead to the similarity calculation between corresponding characteristics in both profiles. Thus, we aim to find characteristics of a patient and group exercise sessions which can build up the new patient information model.

It seems obvious that different properties in the user profile should have different weights. Simply speaking, a weight of a property should represent the importance of that property. For example, users might understand the importance of the property “address” in various ways which could influence the relevance of matching results. Thus, there is a need to investigate the best performing set of weights for a particular user profile. In the end, this set of weights should lead to a good profile matching results.

There are few strategies to assign weights to properties. The simplest is to treat all properties equally. The opposite is to have a different weight for each property. A more complicated strategy is to classify all properties into few groups where each property in the group has the same weight. Even more interesting way to assign weights is, first, to subdivide properties into groups according to some criteria, and then, to treat each group as a separate sub-model. Thus, we aim to investigate the requirements of matching process to develop a strategy to assign weights to properties in the patient information model.

Various similarity metrics can be used to calculate the similarity among user profiles. Basically, the similarity metric is the function which uses the values of properties in the user profile to assign the number to a particular pair of values. Consider having two values “table” and “chair” of the property “furniture”. For humans it is relatively easy to understand that these two properties are related. Next, consider two values “table” and “desk lamp”. Having both pairs of values in mind, which is more similar in meaning? The answer depends on few important aspects. The first could be what is the context or the requirement of the comparison? Next, what possible values can be assigned to a property? Are there some other restriction or parameters? What similarity metrics could be possible used to calculate the relatedness? Some metrics “understand” values as a set of characters with no meaning. On the contrary, other metrics calculate the similarity between semantic meanings. Having this in mind, we develop a framework to investigate the similarity metrics that can be used for the further use. In this thesis, we apply the framework for the syntactic metrics which can be applied to calculate the similarity based on characters of the values.

1.3 Objectives

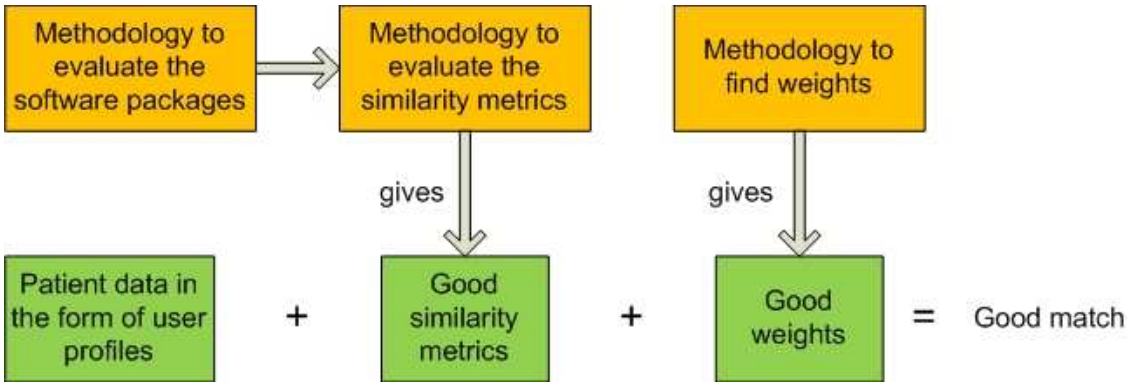
The main objective of this thesis is to design the user information model for patients with CVD, to evaluate syntactic similarity metrics and investigate weights for properties in the user information model. The Patient Comparison System is developed to employ the new patient information model and to find weights. For the application of the framework to evaluate similarity metrics we develop the smaller software application. In this thesis we applied the framework only to the syntactic similarity metrics.

- To design the user profile/user information model for patients with CVD that reflects group exercise session.
- To evaluate syntactic similarity metrics.

- To investigate the weights of properties. Those are used in the new patient information model.

To make it more clear, the figure above shows the high-level application of this thesis. According to the figure, by having the user/patient data, the similarity metrics and weights that can be applied to the user data, we would be able to calculate good matches among users.

Figure 1. The high-level application of the thesis



1.4 Research Questions

In order to reach the objectives of the research, the following research questions are answered in this thesis:

RQ1: What are the characteristics of group exercise sessions? Groups exercise sessions gather people in one place to do physical exercises. Besides that, people experience social communication and supervision. We are interested to find certain characteristics of group exercise sessions.

RQ2: How to define the user information model? The user information model should represent the characteristics of the user or, in other words, identify the user. In this thesis the user information model represents the patients. Thus, in the rest of the thesis the patient information model is an instance of the user information model.

RQ3: How to define the similarity metric. There are different types of similarity metrics which differ in method of application and calculation results.

RQ4: How to find best performing similarity metrics? Because of the variety of similarity metrics, there is a need to evaluate them in order to use them for similarity calculations. We aim to develop and apply a framework which evaluates the similarity metrics.

RQ5: What are good weights for properties in the designed user information model? A matching weight is a measure which is assigned to one characteristic of user and indicates how important the characteristic for the matching process is. We want to explore how different characteristics are important for matching result.

1.5 Research Approach

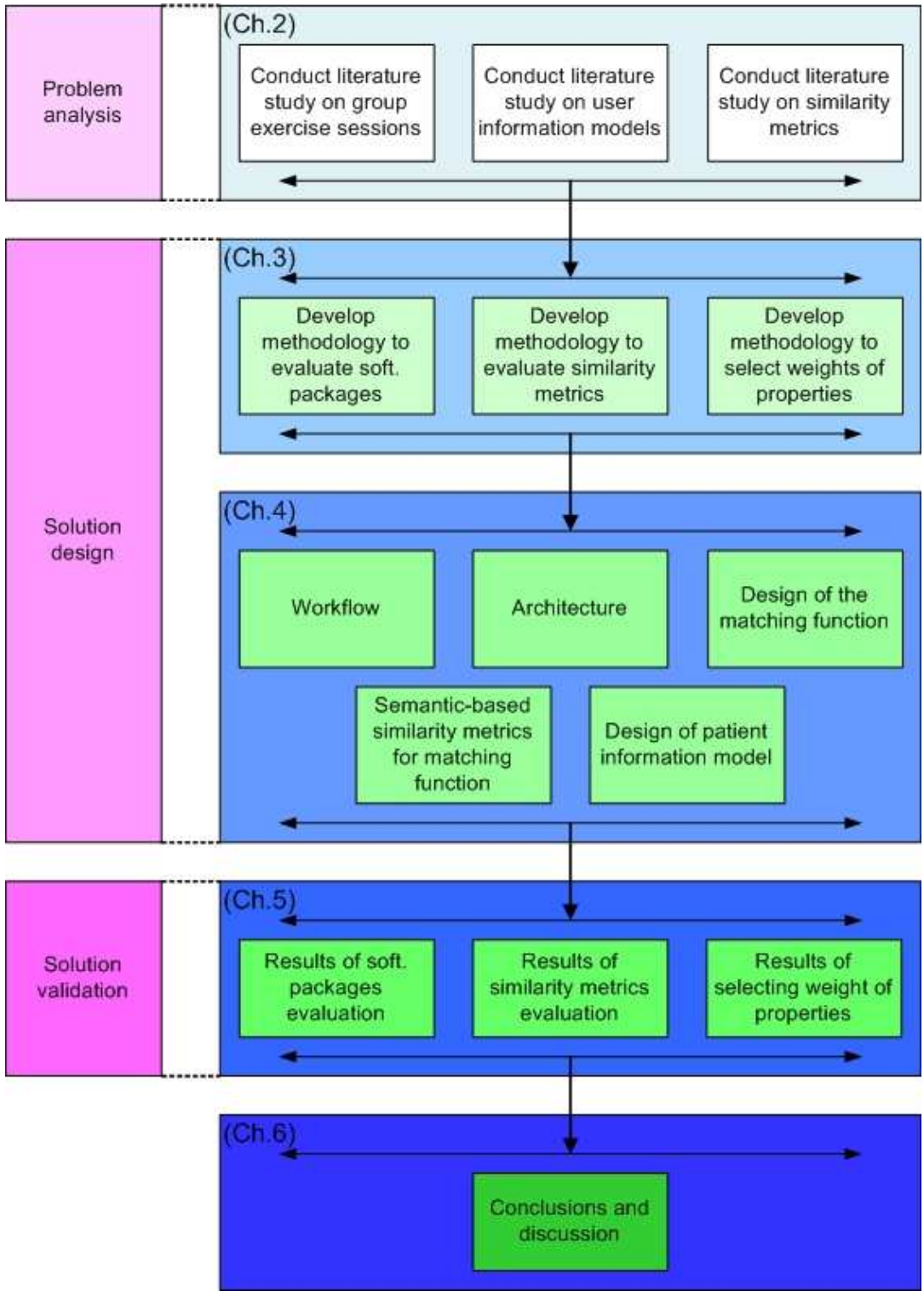
To answer the RQ1, RQ2 and RQ3, we conduct a literature study in the fields of group exercise sessions, CVD, user modeling techniques and existing frameworks. To answer the RQ4, we develop a software application that is used to conduct experiments with a sample of people and evaluate the similarity metrics. To answer the RQ5, we develop a web application which implements our proposed user information model and similarity metrics. This application should be able to calculate similarity between user profiles and give the result to the user.

1.6 Report Structure

To follow the research approach in Figure 1 and to reach research objectives of this study, we structure this research into six chapters:

- *Chapter 2 (Background)* presents the literature findings about important frameworks which are used in user modeling. Also, user modeling techniques are presented. Moreover, we introduce cardiovascular disease and group exercise sessions in terms of benefits for patients. Next, we describe similarity metrics.
- *Chapter 3 (Methodology)* provides means of evaluation of software packages that implements various similarity metrics. Also, we present evaluation method of semantic-based similarity metrics and weights of properties.
- *Chapter 4 (Design of Patient Comparison System)* explains the design choices of PCS system. This chapter goes through system architecture, patient information model, matching function and similarity metrics that are implemented.
- *Chapter 5 (Evaluation of Similarity Metrics and Weights)* presents the results of evaluation by using the method described in the Chapter 3.
- *Chapter 6 (Conclusions and Discussion)* summarizes the thesis and links research questions to results. Moreover, we give some insights for further investigation.

Figure 2. The research approach



2 Background

The main objective of this thesis is to evaluate similarity metrics and find property weights by using PCS which employs user profile for patients with CVD and investigated similarity metrics. In this respect, this chapter presents findings from the literature about four key areas that are important for this study. We present these four areas in separate sections.

The first section is *Frameworks and Classifications* which presents existing means to define business entities like a patient or a caregiver. Also, it provides some classifications and specifications which are important in the health care and will be used throughout this thesis.

The second section is *User Profile Modeling Techniques*. This section presents foundations of user profile modeling techniques which could be helpful in our process of designing the new patient information model.

The third section is *Cardiovascular Disease and Physical Exercises* which briefly present what is CVD. Moreover, in this section, we look at the past studies to investigate how physical exercises can be characterized. Also, we look deeper at the properties of group exercise sessions that show benefits for CVD patients.

The last section is *Similarity Metrics*, which gradually provides the definition of a similarity metric. It also presents the classification of similarity metrics and attempts to characterize two types of metrics, namely syntactic and semantic similarity metrics. These two types are in our interest during this thesis as we aim to evaluate similarity metrics based on the syntactic and semantic structure of comparable values.

2.1 Frameworks and Classifications

We aim to investigate definitions, existing frameworks, which can be used in the process of user information modeling, and classifications of human properties like gender, nationality, race etc. These properties will take place in the new patient information model.

SID – Shared Information Framework

SID is a popular common reference information framework among service providers and vendors. This framework provides guidelines for information description and integration among software applications. It also presents concepts and principles needed to define a shared information model, the entities of the model, as well as the business-oriented UML class models, design-oriented UML class models, and sequence diagrams (Information Framework (SID)).

SID focuses on business entities that are things of interest to the business. In thesis the example of a business entity could be a patient or exercise session. In SID, every business entity is associated by a set of attributes which describe that entity.

OpenSocial

OpenSocial is a framework which enables individuals and organizations communicate across social networks. As the name implies, it is freely available and defines how social networking websites can communicate to each other by a set of standards. The community of OpenSocial claims that these standards do not belong to anyone, thus the improvements are driven by the Internet community (OpenSocial Project).

We are interested in one of the specifications of OpenSocial, namely *social data specification*. It defines all the data objects that are used in OpenSocial. The social data specification consists of *primary social data* and *secondary social data*.

The primary social data defines the entity *Person* which is the recommendation for social networking websites of how to describe a person or a user. The recommendation includes such characteristics as age, body type, interests, etc. The secondary social data defines the entity *Address* which includes properties such as country, postal code, street address, etc. The full list of characteristics of entities *Person* and *Address* is shows in the Appendix I.

FOAF – Friend of a Friend

FOAF provides a specification of how to create machine-readable files which represent people in terms of personal information, links with other people, and what people do. The specification aims to link people by using the Internet (Friend of a Friend Project).

FOAF collects a variety of terms; some describe people, some groups, and some documents. One term is called a *Person.* Person is a subclass of an *Agent*, and the Agent is a parent class for a Person, Organization, and Group. In this thesis we are interested in the ways of characterizing a person, thus we look deeper at the definition of an entity Person in FOAF. Because the entity Person is a subclass of Agent it consists of all properties which can be found in the Agent.

HL7 - Health Level Seven International

HL7 provides a comprehensive framework and related standards for the exchange, integration, sharing, and retrieval of electronic health information. The area of support of HL7 consists of clinical practice, management, delivery and evaluation of health services. More precisely, HL7 aims to create widely used standards in healthcare by improving care delivery, workflow optimization, and enhancement of knowledge transfer among stakeholders.

HL7 also provides, what they call, the vocabulary of concept domains. It is a classification of the encoded information which can be used in the data storage or message transfer. Every concept

domain has a set of values. For example, there is a concept domain *Race* and, according to HL7, it has a minimum set of five values: *American Indian or Alaska Native, Asian, Black or African American, Native Hawaiian or Other Pacific Islander*, and *White*. Another example could be concept domain *AdministrativeGender* which has values of *Female*, *Male*, or *Undifferentiated* (Health Level 7 International).

In the rest of this thesis we will use some of these concept domains in our patient information model.

2.2 User Profile Modeling Techniques

The creation of a user profile and its representation requires the process of user information modeling. In this respect, user profiles are instances of user information model.

Kobsa (1993) proposes a classification of user modeling approaches, namely: (*i*) *User knowledge* approach, (*ii*) *User plans*, and (*iii*) *User preferences*. The first approach looks at the background knowledge of a user by subdividing user groups into subgroups according to the user's key characteristics. The *User plan* approach looks at the sequences of user actions to achieve a certain goal. Finally, the *User preference* approach is useful when there is need to model user profile according to the information needs of users and their preferences.

E. Rich (1983) argues that a user model can describe a huge variety of information. Thus, author introduces the space of user models which classifies them. This classification has three dimensions. The first dimension distinguishes canonical and individual models (one model for everyone versus many different models for everyone). The second dimension classifies explicit and implicit models (models designed by system designer or user versus designed by a system itself). The last dimension characterizes long-term and short-term models (models based on long-term facts versus short-term facts about the user). Furthermore, techniques are introduced which can be used to build user models.

Similarly, Amato and Straccia (1999) investigate non-generic behavior of users to satisfy information needs. There are two steps in the process of user modeling: the *what* dimension, and the *how* dimension. Authors address the goal of the information provider who is to provide the user with the right information, at the right time, through the right means. The representation of the user can be classified in five categories: individual data, gathering data, the delivering data, the actions data, and the security data. As an example, authors present the user profile for the digital libraries and two possible architectural solutions to cope with such profiles.

The presented user modeling techniques concentrate of non-homogenous user modeling to better express user needs and context. There is a high variety of static information that can be classified but there is also a situation-aware or information needs-aware data that has to be expressed by user models. In this study, there is a need twofold user model: canonical and

context specific. On one hand, the user model has to identify humans, and on the other hand, it has to reflect the needs of CVD patients.

For the rest of this study we define user/patient information model as:

The patient information model is a set of personal characteristics that describe the patient and a set of preferences that a patient posses. The preferences describe a group exercise session in which a patient would like to participate.

2.3 Cardiovascular Disease and Physical Exercises

The CVD refer to any king of disease that involves heart or blood vessels and is the main reason of death in the world. There are various ways how to treat patients with CVDs depending on the condition and disease type. Nevertheless, physical exercises remain one of the cornerstones of CVDs rehabilitation (Casillas, Gremeaux, Damak, Feki and Perennou, 2007).

Doing physical exercises became a recommendation as it is a highly cost-effective precaution for the chronic patients (Corra et al., 2005). The exercise sessions should be individualized for every patient as the symptoms, health conditions and ways of treating could differ. In this respect, patients may have preferences for doing exercises.

Prosser, Carson, and Phillips (1985) present long-term effects of doing physical exercises. They conducted a hospital exercise program with selected patients. Exercises lasted about 45 min, they were held twice a week, and had five level of exercise. The result of the experiment shows the correlation between patients who experienced regular exercises and improved cardiovascular and general health status. One of the features of exercise session was that patients enjoyed group activities, which provide supervision, support, and encouragement. The study also presents the reasons why some patient found it difficult to continue the exercise sessions. Reasons include inconvenience, medical reasons, and lack of facilities.

Comparably, Williams and Lord (1997) investigates the effects of group exercise on physiological function, cognition and mood of patients. Results show that patients showed significant improvements in reaction time, strength, and memory span. The group exercises differed in type (walking, cycling, etc.) and in hours that were spend to do exercises.

Casillas et al. (2007) reviews the supervised therapy for patients with CVD. This therapy shows significant improvement in mortality rate, physical capacity, reintegration and overall quality of life. Authors claim that exercise programs should be personalized and correspond to the wide range of diverse situations. However, exercises sessions can be held in groups in an easy to reach venue. The therapy consisted of various types of exercises and was different in terms of frequency, intensity, and duration of the exercises. The types of exercises included: training such as use of bicycle, treadmill, stepping or rowing machines; resistive training such as elastic cables or pulley systems; and muscular electro stimulation.

Mutrie et al. (2007) analyses the functional and psychological benefits of supervised group exercise session for breast cancer patients. The outcome of this study showed improved quality of life in terms of physical, functional, social, and emotional. The exercise groups were held in various times of the day. Moreover, they included various types of exercises such as walking, cycling, aerobics, muscle strengthening exercises, etc.

To sum up the past studies, group exercise sessions have following characteristics:

- They are supervised.
- They should be personalized depending on the personal situation.
- They can be of various types.
- They differ in intensity.
- They differ in frequency.
- They differ in intensity.
- They are held in locations close to patients.

For the rest of this study we define group exercise session as:

The group exercise session is supervised and meets the preferences of an individual patient. The type of an exercise, intensity, frequency, and duration of the session are main characteristics which are driven by patients.

2.4 Similarity Metrics

This section step by step presents what are similarity metrics and how they are classified. Furthermore, syntactic and semantic similarity metrics are presented. We also describe the additional resource *WordNet* which we will use to calculate semantic similarity.

Vector of properties

The user information model can be composed of a set of business entities that have certain properties assigned to them. For example, consider entities such as address, working information and study information. All of these entities can be describes by set of properties. For example, working information can have properties like “office room number” or “working hours”. Moreover, every property can have a value. For example, the value of property “office room number” could be “R-101”. In this study, we will consider such set of properties having mathematical form of vector. Moreover, every instance on an entity we will consider as an object.

For example, there are two objects *e* and *e'* which are characterized by vector of values. If *val* is the value and *t* is the number of values, the following vectors may be defined for *e* and *e'* (Salton and McGill, 1984):

$$e = (val_{e1}, val_{e2}, \ldots, val_{et})$$
$$e' = (val_{e'1}, val_{e'2}, \ldots, val_{e't}) \quad 1)$$

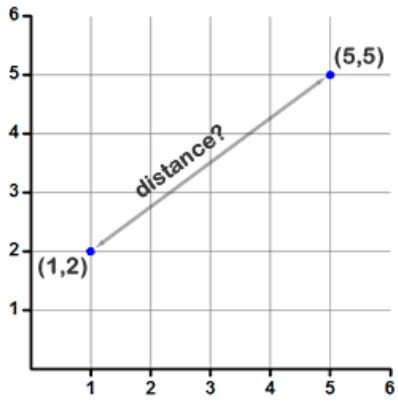
The distance function

The distance function maps a pair of vectors *e* and *e'* to a real number *r*. The higher value of *r* indicates the greater distance between *e* and *e'*. The distance function *d* is required to satisfy the following condition (Distance Function):

- $d(e, e') \geq 0$ (non-negativity)
- $d(e, e') = 0$ when $e=e'$, otherwise $d(e, e') > 0$
- $d(e, e') = d(e', e)$ (symmetry)
- $d(e, e'') \leq d(e, e') + d(e', e'')$ (triangle inequality)

The Figure 2 shows graphical representation of distance between two two-dimensional vectors.

Figure 3. The distance between two points. Adopted from (Distance between two points)



Simply speaking, we aim to investigate what is the distance between two objects (two user profiles) which are characterized by a number of values.

The similarity function

The similarity function is analogous to the distance function because the larger values indicate the higher similarity (Cohen, Ravikumar and Fienberg, 2003). In this respect, the similarity function *sim* equals to:

$$sim = \frac{1}{d} \quad 2)$$

Scientists call similarity functions in different ways, however, majority of them define them in quite the same way. Thus, for the rest of this study, we use names similarity function, similarity technique, and similarity metric interchangeably depending on which interpretation is most natural. Moreover, we consider the problem of similarity only between vectors that have the same set of properties but might differ in values. For example, previously mentioned vectors *e* and *e'* have equal definition but might store various values.

The similarity between earlier defined vectors is a 3-tuple:

$$<e, e', sim> \quad 3)$$

Where *e* and *e'* are the two vectors and the *sim* is the number which indicates the similarity between the two vectors. Based on Ehrig and Sure (2004), we adopt the definition of normalized similarity degree which is a real number in the interval from 0 to 1. The greater the *sim* the more similar are *e* and *e'*.

There are functions that are widely used in the vector algebra and are important in the scope of this thesis. Salton and McGill (1984) list few of them.

The following function is the dot product between *e* and *e'*. It is the sum of an element-by-element multiplication.

$$\sum_{i=1}^{n} e_i \times e_i' \quad 4)$$

Consider each property can be weighted to represent its importance to a given object (Salton and McGill, 1984). Then the weight property vector of *e* is:

$$w = (w_{e1}, w_{e2}, \ldots, w_{et}) \quad 5)$$

The sum of weights of all properties included in vector is the following function

$$\sum_{i=1}^{n} w_i \quad 6)$$

Accordingly, the dot product between *e* and *w* is:

$$\sum_{i=1}^{n} e_i \times w_i \qquad 7)$$

Functions above are not normalized and do not have an upper limit. Thus, the next function is normalized that always gives a convenient result in the range from 0 to 1.

$$\frac{\sum_{i=1}^{n} e_i \times w_i}{\sum_{i=1}^{n} w_i} \qquad 8)$$

Similarity between properties

So far we described similarity between two vectors which have a number of dimensions. As we mentioned before, every vector can be an instance of a business entity. For example, a person can be characterized by a vector *{name, age, height}* which includes properties with different data types and meanings. Then the problem of similarity between two vectors of entity person is more complex that between two numerical vectors. In such case, the similarity is the sum of similarities between each of the corresponding properties in two vectors. Following the example above, the similarity *sim* between two person *p* and *p'* would be:

$$sim(p, p') = sim(name, name') + sim(age, age') + sim(height, height') \qquad 9)$$

In this thesis we will consider that the similarity between two vectors (objects) is the sum of similarities between each of the corresponding properties in these vectors. With respect to this, we will use different similarity metrics to calculate similarity between properties which differ in representation and meaning.

Classification of similarity metrics

Similarity metrics can be classified along many dimensions. In this thesis, we adopt the classification by Euzenat and Shvaiko (2007). Authors classify metrics according to the input of the metric, the characteristics of the calculation process, and the output of the metric. Ehrig and Sure (2004) provide the classification of elementary similarity techniques with two higher-level classifications. These two classifications are: (*i*) *Granularity/input interpretation* classification, and (*ii*) *Kind of input* classification. The simplified classification model is shown in Figure 2.

The *Granularity/Input Interpretation* classification is mainly based on the matcher granularity, i.e., element- or structure-level. The element-level matching and structure-level matching techniques distinguish between computation of correspondences by analyzing entities or

instances of these entities in isolation or in a structure of them (Ehrig and Sure, 2004; Kang and Naughton, 2003). The *Kind of Input* classification is structured depending on what kind of data the matching technique is based on: strings (terminological), structure (structural), models (semantics) or data instances (extensional). The two first ones are found in the ontology descriptions. The third one requires some semantic interpretation of the ontology and usually uses some semantically compliant reasoned to deduce the correspondences. The last one constitutes the actual population of ontology (Ehrig and Sure, 2004).

Chapter 5 gives a more detailed description of vector similarity metrics that are used in the proposed matching process of this thesis.

We will use two types of similarity metrics, namely, syntactic-based and semantic-based metrics.

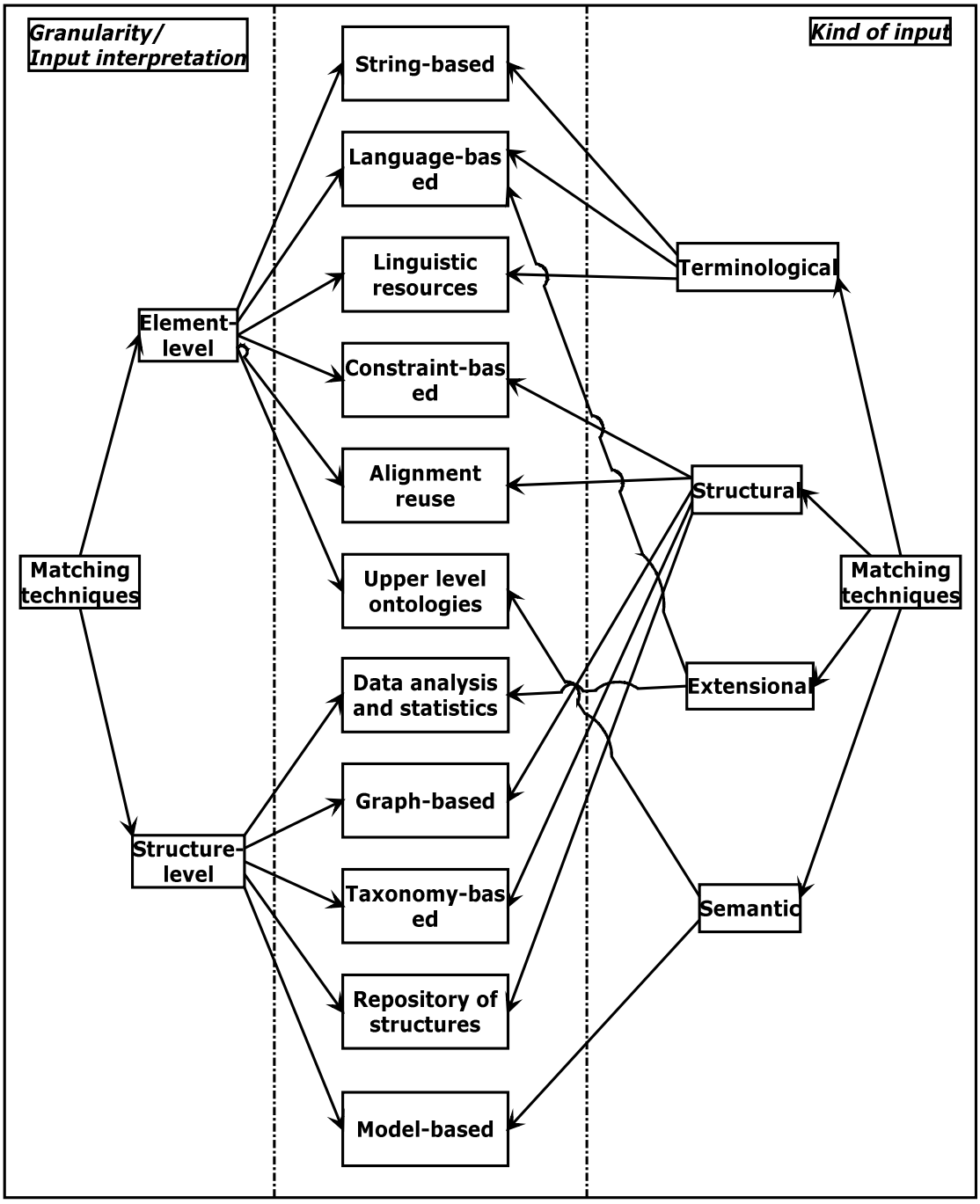
Syntactic-based metrics

We consider string-based (see Figure 3) and syntactic-based metrics to represent the same type of metrics. They calculate similarity between two strings of sequences of letters in an alphabet. Euzenat and Shvaiko (2007) say that syntactic-based metrics "are typically based on the following intuition: the more similar the strings, the more likely they are to denote the same concepts." Basically, syntactic-based metrics are distance functions. One type of distance functions is edit distance (Cohen, Ravikumar and Fienberg, 2003). According to Euzenat and Shvaiko (2007), there exists a sequence of operations that transforms one string into another. Then the distance is cost of sequence of operations which transforms these strings. These operations include insertion, deletion, and substitution. Every of these operations are assigned a cost and the distance between two strings is the sum of the costs. Many mathematical representations and descriptions of most widely used edit distances can be found in Appendix F.

Based on the literature study, we define the syntactic-based similarity metric as:

The syntactic-based similarity metric is an edit function between two strings of characters. It follows the requirements of distance function and is normalized.

Figure 4. Classification of matching techniques



Semantic-based metrics

Based on the classification in the Figure 3, the semantic-based metrics are called linguistic resources. They use additional resources such as thesauri or lexicons to calculate similarity between words. Then the similarity depends on the semantic relations between two words (Euzenat and Shvaiko, 2007).

In scientific literature three different terms, which are closely related to the semantic similarity, are used sometimes interchangeably: semantic relatedness, similarity, and semantic distance (Budanitsky and Hirst, 2006). Sometimes authors take the semantic similarity as a special "case" of semantic relatedness. On the other hand, the semantic distance is the opposite of the semantic relatedness. To illustrate this, authors say that "antonymous concepts are dissimilar and hence distant in one sense, and yet are strongly related semantically and hence close in the other sense" (Budanitsky and Hirst, 2006). For example, "black" and "white" mean two opposite colors; however, they are both colors and are instances of the same concept "color".

In this study, we adopt the notion of the semantic relatedness; however, algorithms of semantic similarity or dissimilarity (as opposite) are widely used in the scope of term semantic relatedness. Thus, these terms will be used in the following sections depending on the context.

Every word in the language can have several meanings. For example, the word *car* can be interpreted differently. One meaning is a road vehicle with an engine and another is a part of the train. These meaning are usually called senses of the word. Because every word can have few meanings, the relatedness between them is the maximum relatedness among every pair of meanings. Consider two words w_1 and w_2, then the relatedness *rel* is defines as:

$$rel(w_1, w_2) = max\big(rel(c_1, c_2)\big)$$

and

$$c_1 \epsilon s(w_1), c_2 \epsilon s(w_2) \qquad 10)$$

Where $s(w_1)$ and $s(w_2)$ are the sets of senses of w_1 and w_2 (Resnik, 1995).

To make it clearer, following is the simplified example of semantic relatedness calculation.
Having two concepts *table* and *chair*, we can make a table of relatedness:

Table 1. The table of relatedness

	Senses of *chair*	
	"A seat for one person"	"The position of professor"

Senses of *table*	"A set of data arranged in rows and columns"	0.1	0.2
	"A piece of furniture"	0.7	0

Numbers in the table represent relatedness between two pairs of senses. Then applying the Formula 10, the relatedness would be the maximum number in the table.

The difference between semantic-based metrics is in the way they calculate the similarity between all possible pairs of senses. Thus, with every other metric the table above would consist of different relatedness numbers.

WordNet

Among many other approaches, semantic-based metrics use WordNet as a resource to calculate the relatedness. Because of its popularity, WordNet can be considered as de facto tool for identification the meaning of words in computational context (Navigli, 2009). WordNet is a large lexical database of English language. It stores nouns, verbs, adjectives and adverbs in grouped sets of synonyms that are called synsets. Synsets are interlinked by means of conceptual-semantic and lexical relations. WordNet is also freely available for download. Its structure makes it a useful tool for computational linguistics and natural language processing (About WordNet).

Basically, WordNet is a lexical database that is structured as a semantic network (Banerjee and Pedersen, 2002) and has the following semantic relations (Miller, 1995):

- *Synonymy* is WordNet's basic relation, because WordNet uses sets of synonyms (synsets) to represent word senses. We can view a synset as a set of word senses all expressing (approximately) the same meaning. Synonymy is a symmetric relation between word forms. Antonymy (opposing-name) is also a symmetric semantic relation between word forms, especially important in organizing the meanings of adjectives and adverbs.
- *Hyponymy* (sub-name) and its inverse, hypernymy (super-name), are transitive relations between synsets. Because there is usually only one hypernym, this semantic relation organizes the meanings of nouns into a hierarchical structure.
- *Meronymy* (part-name) and its inverse, holonymy (whole-name), are complex semantic relations. WordNet distinguishes component parts, substantive parts, and member parts.
- *Troponymy* (manner-name) is for verbs what hyponymy is for nouns, although the resulting hierarchies are much shallower.
- *Entailment* relations between verbs are also coded in WordNet.

As an example of synonymy, following are the meanings/senses of the word "chair" which includes five senses as a noun and two as a verb:

- Noun

- Sense 1: {**chair**} (a seat for one person, with a support for the back). E.g. "he put his coat over the back of the chair and sat down"
- Sense 2: {professorship, **chair**} (the position of professor). E.g. "he was awarded an endowed chair in economics"
- Sense 3: {president, chairman, chairwoman, **chair**, chairperson} (the officer who presides at the meetings of an organization). E.g. "address your remarks to the chairperson"
- Sense 4: {electric chair, **chair**, death chair, hot seat} (an instrument of execution by electrocution; resembles an ordinary seat for one person). E.g. "the murderer was sentenced to die in the chair"
- Sense 5: {**chair**} (a particular seat in an orchestra). E.g. "he is second chair violin"

- Verb
  - Sense 1: {**chair**, chairman} (act or preside as chair, as of an academic department in a university) "She chaired the department for many years"
  - Sense 2: {moderate, **chair**, lead} (preside over) "John moderated the discussion"

For the rest of this thesis, we will consider that semantic-based metrics use WordNet as a linguistic resource to calculate semantic relatedness between two words.

To sum up this section we define the semantic-based similarity metric as:

The syntactic-based similarity metric calculates the relatedness between two words. The metric uses the WordNet lexical database as a knowledge resource and is normalized.

2.5 Chapter Summary

The key points from this chapter are as follows:

- There are various frameworks that present ideas how to build and use user information model. They differ in the context of use and goal.
- We will investigate two types of similarity metrics, namely, syntactic-based and semantic-based similarity metrics.
- Syntactic-based similarity metrics calculates similarity based on sequence of characters in the word. We consider syntactic-based metrics as edit functions which calculate how many insertions, deletions, and substitutions of characters are needed to convert one string to another.
- The semantic-based similarity metrics calculate similarity based on linguistic relations between words. The database of such relations is provided by WordNet – a popular lexical database of English language.

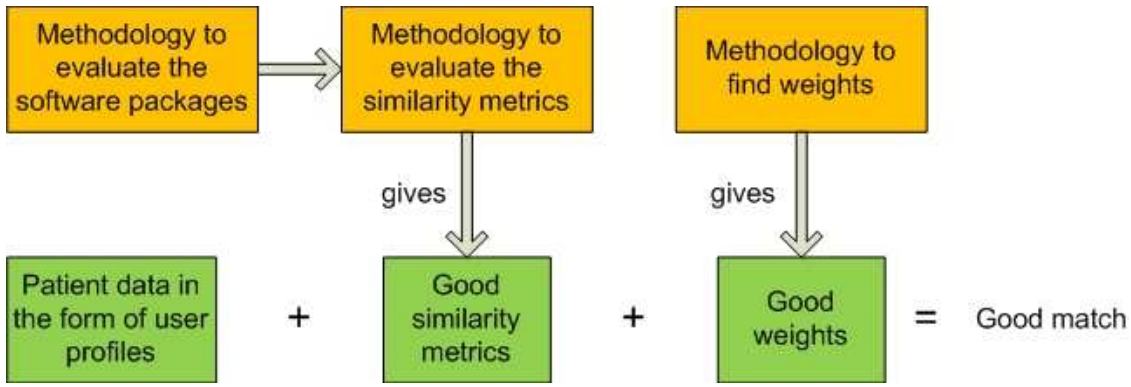
3 Methodology

This chapter focuses on research questions RQ4 and RQ5. Questions RQ1, RQ2, and RQ3 were answered in Chapter 2 by providing literature findings.

This chapter presents the methodology to evaluate software packages and the methodology to evaluate similarity metrics. We explain how we applied both methodologies in order to conduct experiments to assess syntactic similarity metrics. Lastly, we present the method to investigate weights of properties in the patient information model.

This chapter recalls the figure that we presented in Chapter 1 (see figure below). With respect to it, this chapter presents yellow rectangles, namely: *the methodology to evaluate software packages, the methodology to evaluate the similarity metrics, and the methodology to find weights*.

Figure 5. The high-level application of the thesis



3.1 The Methodology to Evaluate Software Packages

The goal of this methodology is to attain software which would be used in the process of similarity metrics evaluation. The software can be attained out of few competing software packages. This methodology consists of three steps:

Step 1 (Finding the software): in this step one has to decide how the software packages with implemented similarity metrics will be obtained from sources such as the internet. We make an assumption that there will be more than one software packages to choose from. The goal of this step is to find as much packages as possible in order to increase the chances to select the one or more that meets the selection requirements.

Step 2 (Trying-out the software): at this step, the found software packages are tried-out. By trying-out we mean the attempt to use it according to the documentation. This step might require additional tools such as Java environment or other that are related directly to the particular software. This step is also a mean to scope the list of software packages but that is not its primary role. For example, naturally some problems might occur during the trying-out that would limit the use of a particular package.

Step 3 (Evaluation to scope the analysis): this step aims to compare the software packages in order to narrow the list of them or to select one or more packages. Various techniques can be used to compare the packages. One can be the technical analysis or efficiency of the package.

This step can be iterative which means that with every repetition other means of evaluation can be applied. The goal of this phase is to attain software package(s) with implemented similarity metrics.

This methodology cannot be applied if the software package is developed in-house. In that case and with respect to this thesis, we could benefit from adjusting the software to the specific needs but that might take more time. In case of using the software from other sources one could benefit from having the complete solution. On the other hand, it can lack the documentation or can be poorly designed. The solution to choose between these options can be led by few circumstances such as the availability of the software sources, the time pressure, the possibility and knowledge to evaluate between few software packages, one's programming experience, etc.

3.2 The Methodology to Evaluate the Similarity Metrics

The goal of this methodology is to select one or more similarity metrics that can be used according to the requirements. Like the methodology in Section 3.1 it has three steps:

Step 1 (Decision how to investigate the similarity metrics): in this step one should decide how to investigate the similarity metrics in the software package. Various questions can be asked in this step in order to make a decision. Some of them can be:

- How many metrics are needed?
- What type of data the metrics have to compare?
- Are there any restrictions about the data?
- Are there any restrictions with regards to efficiency of the similarity calculation?
- What format of the calculation result is required?

This step should contribute in the definite action plan which aims to investigate the similarity metrics.

Step 2 (Conducting the experiments): this step performs the decision described in the Step 1. The outcome of this step is the statistical data that can be analyzed in order to choose one or more similarity metrics.

Step 3 (Conclusions and selection): based on the statistical data in the Step 2, in this step one has to summarize the results of experiments and select the similarity metric(s).

Few remarks about the methodology to evaluate the similarity metrics:

- Step 2 and Step 3 might require additional tools for conducting the experiments or for analyzing the statistical data. By tools we mean additional software applications that could help to achieve goals of the step. Such software applications can be either developed or attained from other sources.
- The list of ways to perform experiments or analyze the data is out of the scope of this thesis because it depends on the research questions that needed to be answered. Our application of this methodology is presented in the following section.

3.3 The Application of Both Methodologies

This section follows both methodologies which we presented in the previous two sections. Here we elaborate on how we applied these methodologies to investigate the software packages and syntactic similarity metrics.

The combined framework is shown in the Figure 6. According to it first we applied the methodology to evaluate the software packages and then we proceeded with the methodology to evaluate the similarity metrics.

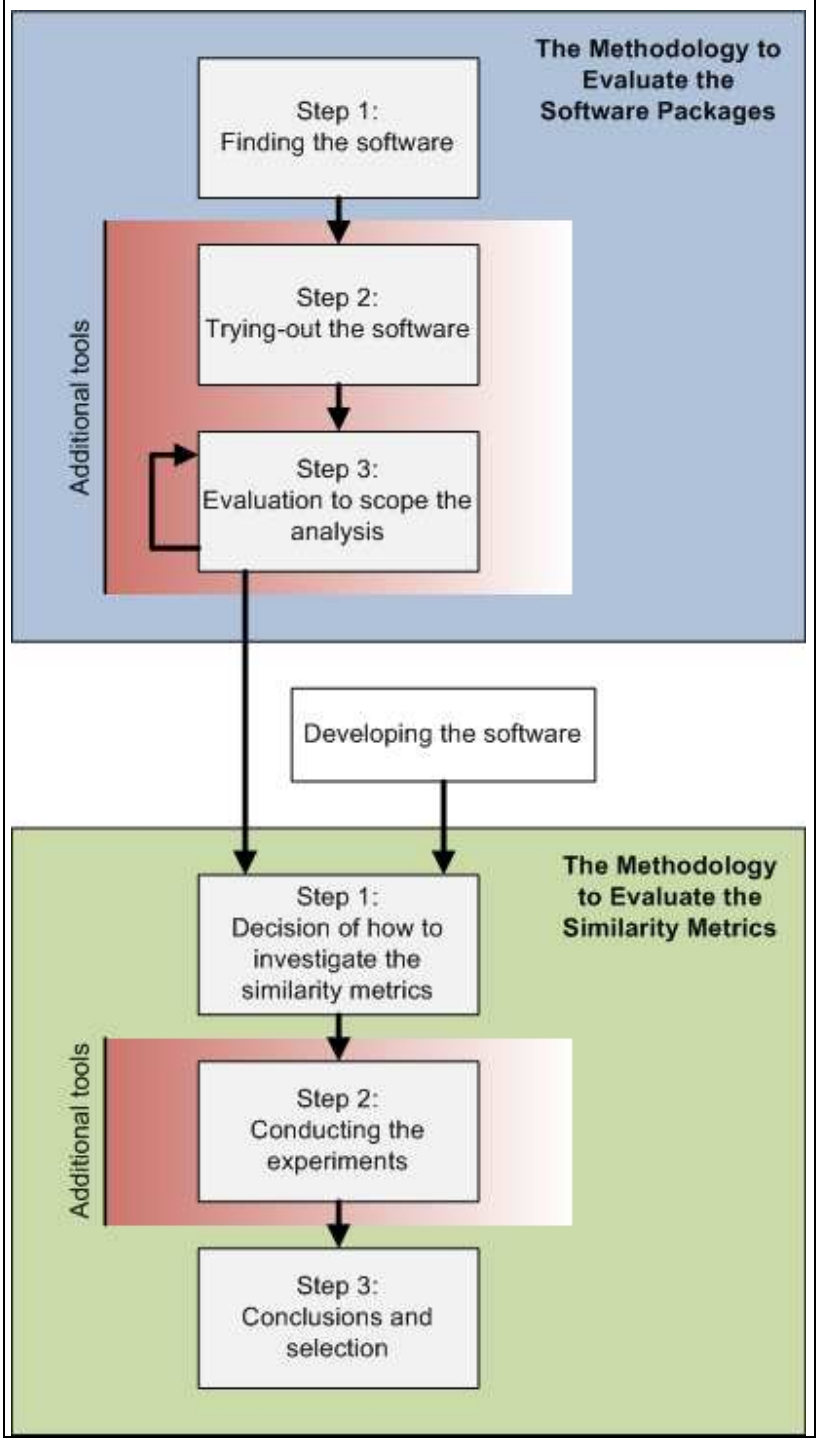
Application of methodology to evaluate software packages

Step 1 (Finding the software): our aim was not to develop implementations of similarity metrics but, instead, use existing implementations that are freely available. We used the internet to find freely available software. Developing the software was out of scope of this thesis.

Our analysis of software started with two steps which are explained below. With every step we expected to narrow the scope of analysis which finally would lead to one software package that we could use in the final assessment.

Step 2 (Trying-out the software): in this step we downloaded the packages and tried to use them according to the documentation. To try out these packages we developed a testing software application which we called *Nile.* The user interface and rough architecture of Nile is shown in the Appendix E.

Figure 6. The methodology to evaluate the similarity metrics



By using Nile we aimed to get some insights for further investigation. For example, it might happen that some packages were outdated, poorly documented or designed which could lead to issues related to the further use.

The process of trying out the packages was quite straightforward. Nile was able to take input of two strings which were compared by every package separately. We used different inputs and monitored any software error that could cause any issue for further use of the packages.

Because we expected to find more that one software packages and the goal was to use just one of them the following steps describe the actions that helped to select just one package.

Step 3 (Evaluation to scope the analysis): it was meaningful to investigate the differences among packages in terms of metrics that have equal names. For example, some of the packages have the metric with a name of *Jaro*. Our goal in this step was to know if metrics with equal name such as Jaro provides equal calculations in all packages.

In this step we also used application Nile. The following section describes the experiments with Nile to investigate metrics with equal names.

Data and methodology of experiment in the Step 3:

We made a list of metrics with equal names that are implemented in more than one package. Further, only metrics in this list were investigated.

Roughly speaking, the output of calculation of similarity metric depends on two criteria: input strings and additional parameters that are used by a metric. If the metric required additional parameters we tended to use equal parameters with all metrics that have equal names. Thus, the input is the only criteria of investigation in this step.

We made three iterations to analyze the metrics with equal names. Length of input strings was the only difference between iterations because. More precisely, first iteration was conducted by using one word, the second iteration used three words and the last iteration used one full sentence.

For the first iteration we picked 6 words of different length. According to Sojka (1995), the average word length of US English language is 8.93 characters and 74% of all words are of length from 6 to 11 characters. In this respect, we generated 6 random strings with lengths of 6, 7, 8, 9, 10, and 11 characters. To generate the strings we used the generator in (Random String Generator) and the Table 2 shows the generated strings.

Table 2. Randomly generated one word

Length	String
6	‘aqyiqs’

7	‘nanbwcr’
8	‘jndpeowa’
9	‘lylnxehgy’
10	‘rhcqazeqpz’
11	‘mjalxxhdvjc’

For the next iteration, we changed the input to longer strings which consisted of three words. The same generator was used get strings which are listed in the Table 2.

Table 3. Randomly generated three words

Length	String
3 words (19 characters)	’impending roll wish’
3 words (18 characters)	’jolly emphasis fed’
3 words (25 characters)	’guaranteed locking bottom’
3 words (32 characters)	’invited infrastructure replacing’
3 words (16 characters)	‘late hand parsed’
3 words (20 characters)	‘cynical advert breed’

Finally, we compared metrics by changing the input to one full sentence that were generated by sentence generator in (Random Sentence Generator). The input sentences are listed in the Table 3.

Table 4. Randomly generated sentence

Length	String
A sentence (4 words, 24 characters)	‘a potential drip reasons’
A sentence (7 words, 37 characters)	‘beneath a toe bobs a minute circuitry’
A sentence (10 words, 73 characters)	‘the geared throughput invokes the nuisance underneath its arranged rocket’
A sentence (7 words, 42 characters)	‘a graduate tax farms underneath her friend’
A sentence (10 words, 53 characters)	‘a new bomb constrains the tree past the national dish’
A sentence (7 words, 44 characters)	‘the welcome salary skips across the engineer’

As we explained in the methodology in Section 3.1, the Step 3 can have few repetitions. We preceded the analysis with the second iteration because we made an assumption that the first

iteration might give unexpected results. For example, few metrics with equal names produce contradicting calculation. In that case, we would investigate the source code of packages to find out the reasons that leaded to the computational differences. On the other hand, if our assumption would be false then we would skip this step and continue with the evaluation of similarity metrics.

Supposing our assumption was true and few packages calculated not equal similarities with metrics that have equal names. In that case we first checked if the metric name and the source code in the package correspond to the mathematical formula defined by original authors of the metric. Second, we paid attention to additional parameters that are used by some metrics. Third, we analyzed if different metrics interpret the input strings equally. For example, software packages might apply transformations to input strings such as eliminating numbers or taking into account case sensitivity. Fourth, we investigated the data types that are used in the source code.

After two iterations we summarized the results concluded the analysis by selecting one software package that was used for further investigation. This investigation aimed to select one similarity metric that was implemented into Patient Comparison System for syntactic matching between properties of user profiles.

Application of methodology to evaluate similarity metrics

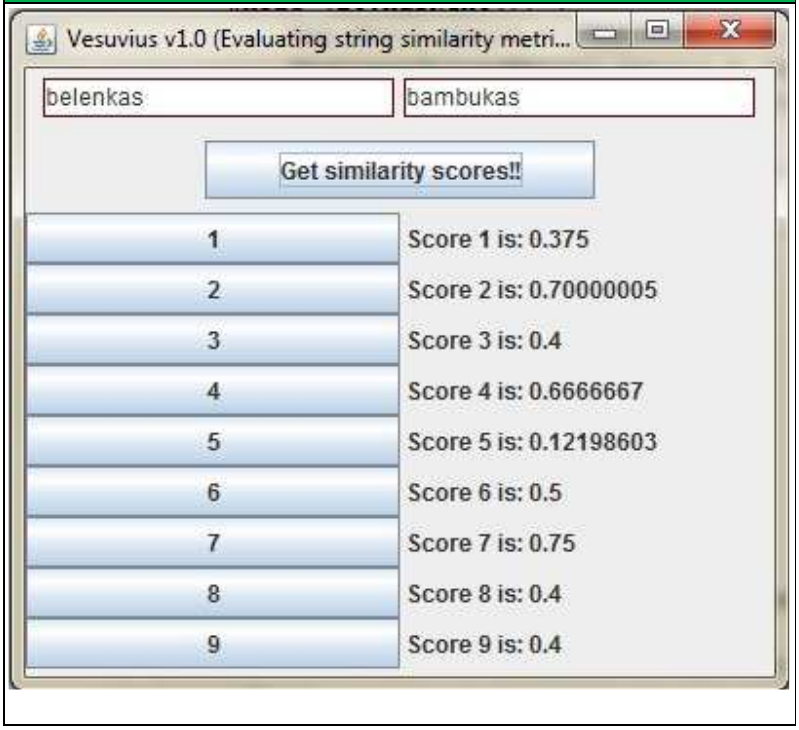
Step 1 (Decision how to investigate the similarity metrics): the goal of this evaluation is to find out how to select the best performing syntactic metric. Thus, at this point we already have selected one software package that we decided to use for this evaluation. Furthermore, we investigated differences between metrics that have equal names. With respect to this, we wanted to leave the list of metrics with equal names which was made in the previous section.

For further investigation of similarity metrics in selected software package we developed a small testing software application which we called *Vesuvius* (see Appendix L for Vesuvius architecture). It aimed to help to analyze the performance of metrics (see Section 5.1 for more details on metrics which were implemented into Vesuvius) by using judgments of humans about the calculated similarities. Figure 7 shows the user interface of Vesuvius. The top row input fields in Vesuvius are used to input two strings and the button “Get similarity scores!!” was used to trigger metrics to calculate similarities between input strings. The calculated similarities were shown in the right side of the application.

During the experiment, a person had to look at similarity values and decide which is the most correct according to him/her. The person had to press the corresponding button on the left which tracked back the metric name and wrote it to the system log file. To keep focus of the person during the experiment, similarity metrics were assigned to numbers and not original names. Every metric was randomly assigned to a different number on the left side of the

application. After each repetition of the experiment metrics were randomly assigned again. In this way, the person was not able to recognize the patterns in the metric calculations.

Figure 7. Vesuvius application



Data for experiment

The experiments were based on two types of input data. The first type was the free-noun-input which allowed people to choose the input data by themselves. There were no restrictions for input data. The second type of input data was a set of nouns in English language (fixed-noun-input). The words in the set were chosen by us. The set consisted of five randomly chosen nouns: *disease, heart, blood, vessel, disability*. It is worth to mention that the words that we selected did not have a need to be of a specific context; however, to follow the problem context of the thesis, we picked words from heath care domain.

Method of experiment

A sample of 10 adults was asked to participate in the experiment by using *Vesuvius*. The participants varied in gender, age (mostly students between 20 and 32 years), and background. There were no participants who had CVD or had a background or experience in the healthcare.

The experiment was conducted in two sessions. They differed in the input data type that was used: free-noun-input and fixed-noun-input.

During the first session participants had to use *Vesuvius* with nouns that they think of. They were briefly explained how to use Vesuvius and how did it calculate the similarity scores. We also explained that the input nouns are compared in terms of syntactic and not semantic matter. In total, every participant was asked to input 10 pairs of nouns for this session. Thus, this session produced one hundred results.

During the second session, each person was asked to use combinations of nouns from the Table 4. The table was filled by generating all possible combinations among five nouns mentioned above. The rest of the experiment was equal to the first session. As the first session, this session resulted in one hundred of results.

Table 5. All combinations for the second session of the experiment

Noun A	Noun B
disease	heart
disease	blood
disease	vessel
disease	disability
heart	blood
heart	vessel
heart	disability
blood	vessel
blood	disability
vessel	disability

The application of the Phase 2 and the Phase 3 will be presented in Chapter 5 where the statistical data from the experiments is analyzed.

3.4 The Methodology to Select Weights of Properties

To reach this point of our research and be able to find weights for properties, we had to be sure that two assumptions were already true. First, patient information model is designed. Second, PCS is developed with required functionality to assess the weights and calculate similarities among user profiles.

The goal of this section is to explain how to find the set/s of weights for properties in our proposed patient information model. The investigation consisted out of five steps.

(Step 1) Building 8 sets of weights: intuitively, to find the best performing set of weights we needed to analyze few of them. In this step we sampled 8 adults to create weight sets for further investigation. We listed all properties of the patient information model and asked every participant to assign a number to every property in a given range. For the purpose of simplicity

we selected the range to be from 1 to 100. For example, there are two properties by names “age” and “gender”. Then every participant had to assign one number to “age” and one number to “gender” which indicated weights of these properties. The sum of weights did not have to be equal to 100 or any other number. In general, the weight numbers meant how one property is more or less important in relation to other.

(Step 2) Running the matching function among all profiles: we triggered the matching function which calculated similarities among all profiles in PCS by using the weight sets from the Step 1 (see next chapter for definition of matching function).

For example, consider there are *n* filled patient profiles in PCS and *m* sets of weights from the Step 1 in this section. The matching function would calculate *n x (n - 1)/2* similarities with one set of weights. With all sets of weights it would produce *n x (n - 1) x m/2* calculations (see Section 4.3 for more details about the matching function).

(Step 3) Let people to assign similarity score between profiles: in this step we aimed to gather human judgments about similarity of every pair of profiles.

Every participant was asked to follow the same workflow of the assessment in this step.

The workflow of the assessment:

The assessment procedure started with showing two profiles for a participant in the user interface of PCS. One of the two profiles was always of the participant because, in such a way, we expected to receive more precise results. The participant had to visually compare the information in the user profiles and assign one number which indicated his/her overall opinion of the similarity between these two profiles. After the assignment PCS showed another pair of profiles and such procedure continued till the participant assigned numbers of similarity of eight corresponding pairs of profiles.

We made an assumption that assigned similarity scores between symmetric pairs of profiles would differ. For example, if we have profiles P1 and P2, then similarity score assigned by P1 and similarity score assigned by P2 might differ. In that case we would have to choose one of the assigned numbers that would participate in the following calculations. The reasoning to choose one of the numbers is out of scope of this thesis. To eliminate this issue we calculated an average of scores of symmetric pairs of profiles.

(Step 4) Compare the similarity scores between matching function and human judgment: the objective of this step was to select one or more sets of weights that correspond to the opinion of the subjects. To achieve this goal we calculated the correlation between the assigned similarity scores by humans and the calculated similarity scores by PCS. The highest correlation scores would mean a strong relationship between the calculated similarities. According to the Yang (2010), the correlation value can vary in the range [-1, 1]. In our case the correlation scores could mean few different things:

- The closer the correlation value to -1 the more negative is the relationship. Basically that would mean that the human opinions about how similar are profiles are highly opposite to the calculated similarities by PCS.
- If the correlation value is close to 0, then the similarities have no relationship at all.
- If the correlation value is close to 1, then both types of similarities have a strong relationship and it is possible to make predictions about the behavior of one similarity when given the other. In our case that would mean that the calculated similarities by PCS correspond to the assigned similarities by humans. Our aim is to select one set of weights with which the calculated similarities have the strongest relationship with the assigned similarity.

(Step 5) Calculate the weights by using the linear regression method: in this step we took a different approach and calculated the weights by using the methods of linear function calculation. Basically speaking we calculated weights that best fits our statistical data from the experiments. To calculate the weights we used two additional software applications: *Microsoft Office Excel* (About Microsoft Office Excel) and *MathWorks Matlab* (About MathWorks Matlab). The formula that we used to calculate the weights in both applications is shown below.

$$SIM_{1,2} = w^1 \times sim^1 + \cdots + w^n \times sim^n$$

Where $SIM_{1,2}$ indicates the similarity between profiles P1 and P2 which was assigned by subjects during the experiments, w^n shows the weight for a particular property, that we wanted to calculate, and sim^n is the similarity between property *n* which was calculated by PCS. For the simplicity we calculated weights only using information for profiles P1 and P2. The calculations of all other 27 combinations of profiles are out of scope of this thesis due to the complexity of gathering all sim^n values from the PCS.

In the Microsoft Office Excel we used the linear regression function LINEST (About Microsoft Office Excel function LINEST) which calculates a straight line that best fits our statistical data by using the "least squares" method.

In the MathWorks Matlab we used the function LINPROG (About MathWorks Matlab function LINPROG) which solves linear function equations. In essence it is similar to LINEST but can satisfy additional constraints for the solution. In our case such constrain is that the weight can be in the range [1:100].

By using two different software applications to calculate the weights we aimed to increase the chance to get a satisfying result. After the calculations were done we compared the calculated weights and the assigned weights to make a conclusion which explains the possibility to calculate weights for every person individually.

3.5 Chapter Summary

The goal of this chapter was to present the methodology to investigate the software packages (see Section 3.1) and the methodology to investigate the similarity metrics (see Section 3.2). The methodology to investigate the software packages aimed to select one package that would be implemented in Patient Comparison System to calculate similarities among user profiles. This chapter also describes the application of both methodologies (see Section 3.3).

Next we present the methodology to select the weights for properties in the patient information model (see Section 3.4). This methodology includes the use of PCS that calculates the similarities among profiles.

We applied the methodology to investigate the similarity metrics only for syntactic metrics. The reasoning for not considering the semantic similarity metrics was the research from Budanitsky and Hirst (Budanitsky and Hirst, 2006). They concluded that *Jiang and Conrath* metric outperformed other metrics during the experiment (see Section 4.4 for more details). This metric was implemented in Patient Comparison System to calculate semantic similarity.

4 Design of Patient Comparison System

This chapter presents the design of Patient Comparison System (PCS). We developed PCS to be able to apply the designed patient information model, implement semantic-based similarity metrics, and to help to analyze the weights for properties in the patient information model. The main objective of PCS is to calculate similarity among user profiles. The main function of PCS is the *matching function* which calculates similarities. The diagram which explains the application of PCS is shown in the Figure 8. Moreover, according to the methodology presented in Chapter 3, PCS takes a role of a additional tool in the Phase 3.

To describe the purpose of PCS we adopt notation by Wieringa (2003) which provides the high-level description of desired system functionality or mission statement. Following is the mission statement for PCS:

Table 6. The mission statement of PCS

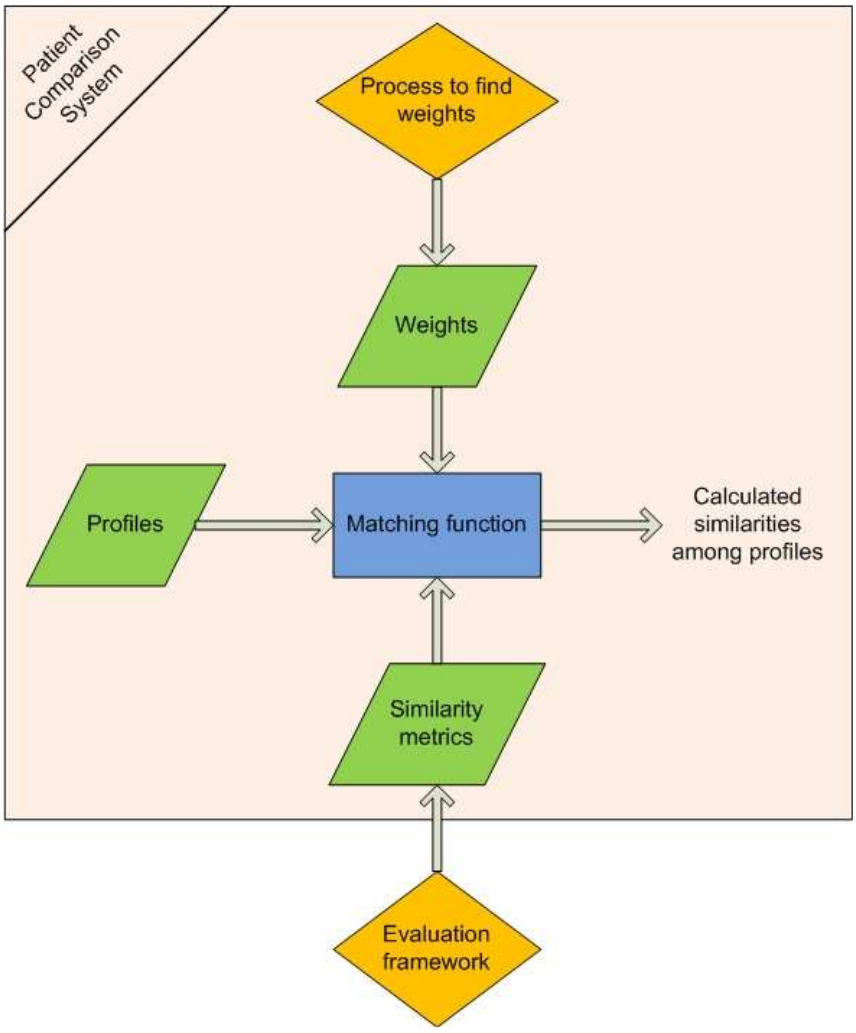
Name	Patient Comparison System
Acronym	PCS
Purpose	To implement the patient information model, support the analysis of weights for that model and be able to calculate the similarities among patient profiles.
Responsibilities	• To implement the patient information model. • To implement semantic-based similarity metrics and WordNet as a resource. • To calculate the similarity between patient profiles. • To provide functionality for analysis of weights.
Exclusions	• The system will not support any other services that might be a part of community for CVD patients. • PCS does not support a high number of profiles due to the simplicity of the architecture • PCS does not implement syntactic-based similarity metrics.

The authors of this thesis have experience in Java programming language; therefore, PCS is based on this programming language and technologies of Web Applications (see Appendix C for the more concrete list of technologies).

4.1 Workflow of use

There are two types of users in PCS who play different roles. First is the user who plays a role of CVD patient. This role is responsible for submitting required information into PCS. The information consists of individual information and the preferences for group exercise sessions. The second role is the caregiver. Caregiver does not have a user profile because his/her main

Figure 8. The application of PCS



responsibility in PCS is to trigger matching function. In addition, caregiver can set the weights for all properties of the user profile.

Both roles have different functions and different user interfaces in the PCS (see the Appendix D).

Workflow of the patient: a patient can log in PCS and see his/her user profile. The patient can modify information in the profile and save it. After saving the modified information will be seen next time the patient logs in. The patient does not see other user profiles or the page of caregiver.

A patient can also assign similarity between two profiles. When the patient starts the procedure of the similarity evaluation, PCS shows two profiles in one page. One of the profiles is the current patient/user and another is changed every time the current patient assigns a similarity measure to a pair of profiles. Such procedure continues till all pairs of profiles between current user profile and other profiles are assigned with a similarity number. This information is saved and later analysed to investigate the weights of properties.

Workflow of the caregiver: the caregiver logs in and sees two main pages. On the first page, he/she can see all the user profiles submitted in PCS. For the purpose of analysis, these profiles can be compared by triggering matching function only for two selected profiles. In the second page, caregiver can submit weights for properties in the user profile.

A caregiver is also able to trigger a matching function between all profiles. Then the matching function calculates similarities between all pairs of profiles in PCS. The result is saved in the system and shown in the user profile.

4.2 Architecture

The Figure 9 shows the context diagram of the PCS. The black lines represent communication channels between entities, namely: the web browser and the profile database. The web browser provides the user interface to the users. The database stores the information of user profiles and information about weights of properties of user profiles.

The PCS is a Java-based web application. Thus, it runs in the web server.

Figure 9. The context diagram of PCS

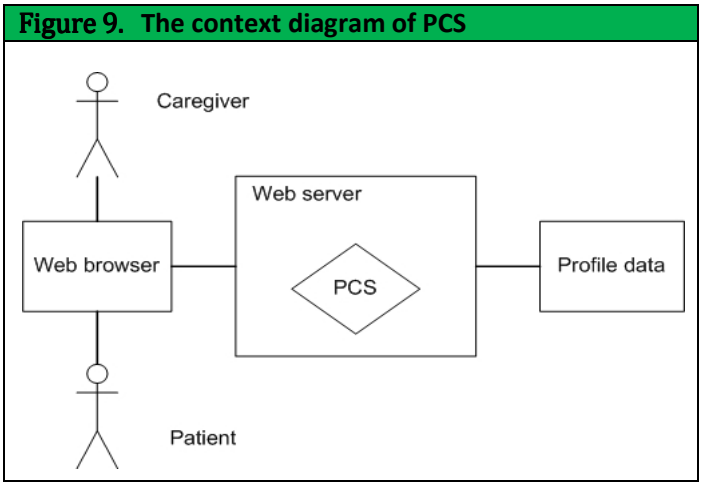
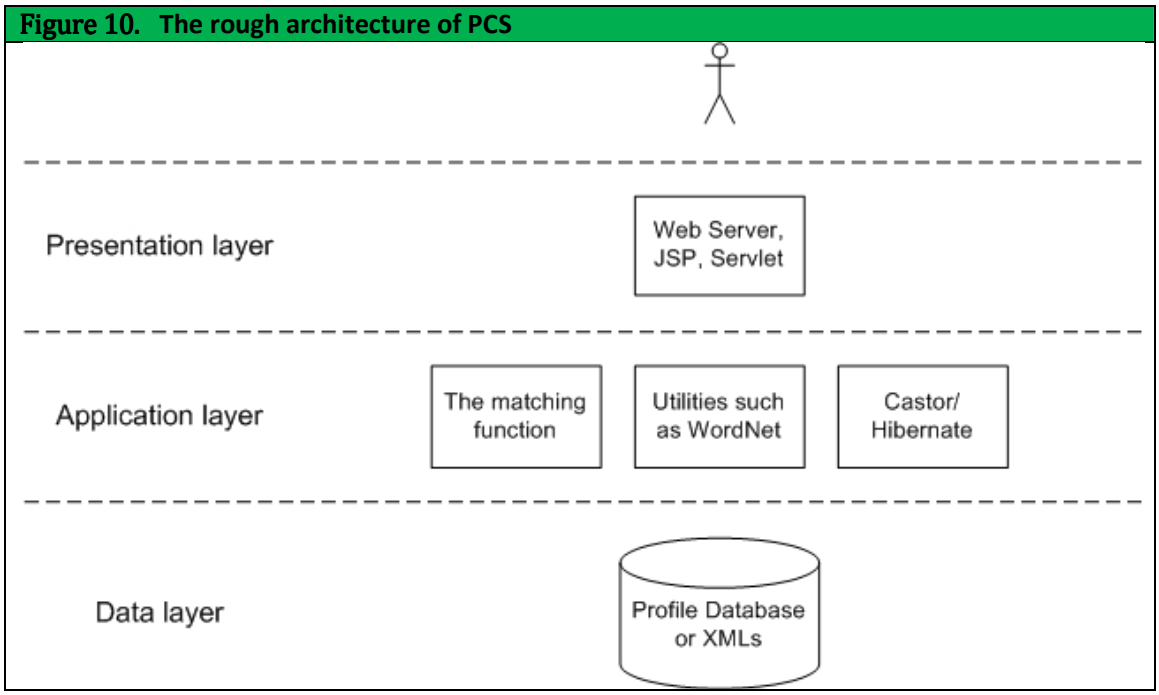


Figure 10. The rough architecture of PCS



Moreover, PCS is based on technologies such as *Apache Tomcat* (Apache Tomcat)*, Hibernate* (Hibernate), *Java EE* (Java Enterprise Edition), *and MySQL* database (MySQL). Our main reasoning to choose the mentioned software was the possibility to get it free, the popularity among developers, and the possibility to deal with objects in Java programming language.

The rough architecture of PCS is shown in the Figure 10. It is based on three layers: the data layer which ir responsible for storing the data, the application layer which performs the business logic of the application, and the presentation layer which is responsible for showing user interfaces and providing input functionality to the user.

4.3 The matching function

The *matching function* is the main function of PCS and it is triggered by caregiver role. The matching function calculates the similarity between values of properties in the user profile. For example, having profiles A and B, it calculates the similarity between values of "age" in A and B. The matching function calculates similarity between corresponding values of properties that are listed below. For example, the matching function calculates the similarity between two values of "Gender" or two values of "Age".

Table 7. The properties for the matching function	
Name of property	Entity in the class diagram
Gender	Individual
Nationality	Individual
Marital status	Individual
Age	Individual
Disabilities	Individual
Likes	Individual
Dislikes	Individual
About me	Individual
Language spoken	Individual
Duration	Exercise preference
Intensity	Exercise preference
Frequency	Exercise preference
Exercise type	Exercise preference

It is worth to mention, that matching function does not calculate similarity between following properties:

Table 8. The properties which are excluded from the matching function	
Name of property	Entity in the class diagram
Given name	Individual
Family name	Individual

We excluded these properties from matching function because they do not represent characteristics of individual or preferences for group exercise sessions. These properties were included in the patient information model because there was a need for user identification in PCS.

We adopted the weighted property-vector approach. Thus, every property that is listed in table 7 is assigned a real number. This number which we call *property weight*, indicates how important is the similarity between the values of this property to the final matching result. The higher the property weight the more important the similarity of this property is. For the convenience, the weight can vary from 1 to 100 so the caregiver who uses PCS must follow this notation.

Assuming we have user profiles *A* and *B* with *i* properties and every value of the property in the profile is defined as v_i . We define similarity between every v_i in *A* and *B* like:

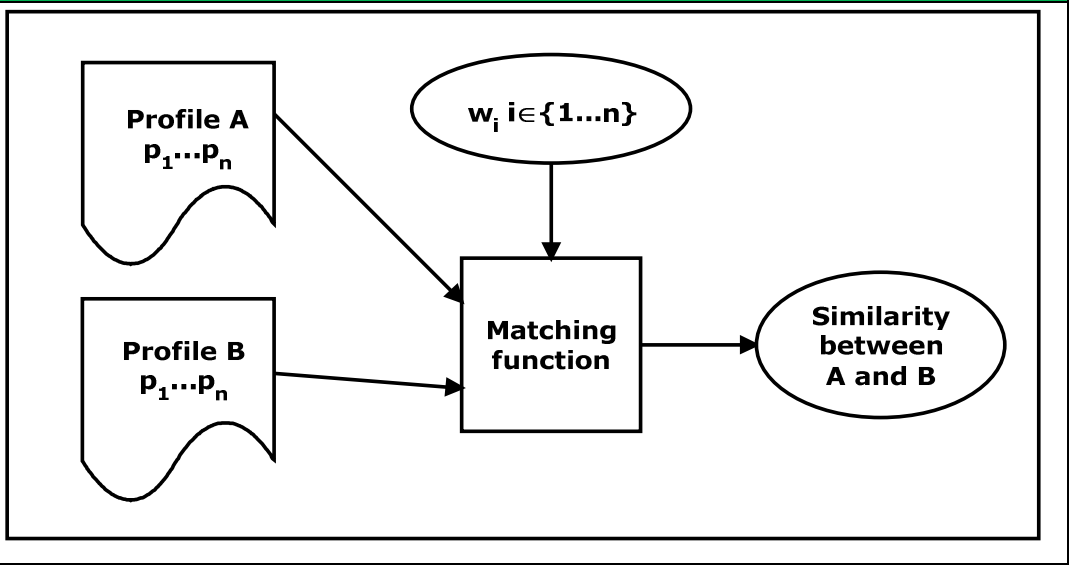
$$sim_i\left(v_i^A, v_i^B\right) \quad 11)$$

Then the matching function calculates the similarity *SIM* between profiles *A* and *B* with the formula

$$SIM(A,B) = \frac{\sum_{i=1}^{n} w_i * sim_i\left(v_i^A, v_i^B\right)}{\sum_{i=1}^{n} w_i} \quad 12)$$

Where w_i indicates weight of v_i. We adopted the normalized representation of the similarity metric. Thus, the formula above gives the result as a real number in the range from 0 to 1. Graphically, the matching function is shown in the Figure 11.

Figure 11. The matching function



In the patient information model we can distinguish a few types of properties that should be compared differently by the matching function. For example, values of “age”, “nationality”, and “disabilities” should be compared in different ways. “Age” is represented by a number, “nationality” is a value from the fixed list of nationalities, and “disabilities” can have a value that can highly vary in meaning. To solve this problem, the matching function uses two methods to calculate the similarity.

Semantic: The first method we call semantic-based method because it calculates similarity based on semantic relationships between input values. More specifically, it finds the maximum similarity between senses of input values by using WordNet database as a resource to calculate

the similarity. We rely on the past studies which investigated the semantic similarity metrics. Thus, we selected *Jiang and Conrath* metric to be implemented into PCS (Budanitsky and Hirst, 2006).

Numeric: this method is used to calculate similarity between two values based on their numeric representation. For example, the property age in our proposed patient information model can have six values: “<30”, “30-39”, “40-49”, “50-59”, “60-69”, and “>69”. Because these values are valid for every user of PCS it is more convenient to save the value not as the string data type but as a number. Then the values representing the same list of option of the age would be: “1”, “2”, “3”, “4”, “5”, and “6”. To make it clearer, consider that a patient saved that he/she is less 30 years old. Then his/her age in PCS would be saved as a number “1”.

Having this notation, it is easy to calculate the similarity between two values which are represented by a number. In this case we proposed that the similarity would be an absolute value of difference between the two values multiplied by a scaling coefficient. The scaling coefficient is needed to calculate the result in the range [0, 1] and it equals to *1/(the number of intervals – 1)*. Following the example above, this coefficient would be 1/(6-1) = 0.2;

It is worth to mention that two values of the same property in two profiles are compared using only one method listed above. We do not consider multiple applications of metrics for one property or combination of metrics for one property. Thus, the following table shows which methods are used for each property in the patient information model.

Table 9. The properties and matching methods for them

Name of property	Entity in the class diagram	Method name
Gender	Individual	Syntactic
Nationality	Individual	Syntactic
Marital status	Individual	Syntactic
Age	Individual	Numeric
Disabilities	Individual	Semantic
Likes	Individual	Semantic
Dislikes	Individual	Semantic
About me	Individual	Semantic
Language spoken	Individual	Syntactic
Duration	Exercise preference	Numeric
Intensity	Exercise preference	Syntactic
Frequency	Exercise preference	Numeric
Exercise type	Exercise preference	Syntactic

To sum up, besides information in the form of user profiles, there are other attributes that can add to the definition of the matching function such as external resources used by the similarity metric, e.g., WordNet. Thus, the matching function is defined as the function *f* which, from a

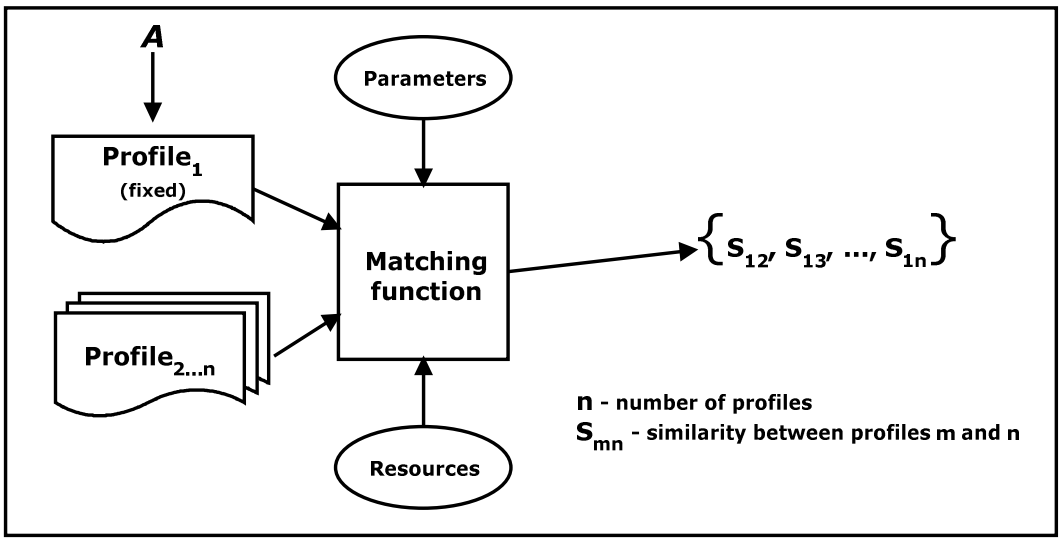
pair of user profiles *P* and *P'*, a set of matching parameters *p*, and a set of external resources *r*, returns the similarity *SIM* between these profiles:

$$SIM = f(P, P', p, r) \quad 13)$$

Example

Suppose there is a profile *A* and the user of this profile wants to know the possible matches of his/her profile with other profiles. Then profile *A* has to be compared with every other profile in the PCS database. The outcome of the matching process is the set of similarities between *A* and every other profile. Such, more complex, matching function is graphically shown in the Figure 7.

Figure 12. The more complex representation of matching function



The process in the figure above is applied for every profile; consequently, the result of the matching is the two dimensional matrix of similarity scores which express all possible similarities between two profiles. This matrix is called similarity matrix. The Table 4 shows an example of such matrix for five possible profiles.

Table 10. The similarity matrix for five profiles

	P_1	P_2	P_3	P_4	P_5
P_1		0.1	0.5	0.3	0.6
P_2	0.1		0.9	0.8	0.2
P_3	0.5	0.9		0.2	0.4
P_4	0.3	0.8	0.2		0.5
P_5	0.6	0.2	0.4	0.5	

P_1, P_2, P_3, P_4, and P_5 indicate user profiles and every cell in the matrix express the similarity between two profiles. The table above is an example and is filled with random values in the range [0, 1] where higher values mean a "more" similar pair of profiles and lower values mean dissimilar pair of profiles. The matrix is *n*n* symmetric because the computations between two equal profiles gives the same result, e.g., in the matrix SIM_{14} and SIM_{41} are equal. Thus, we are interested only in half of results which are indicated by grey color in the matrix.

Following the example above, the matching function definition can be extended for the profile P_1. The improved definition says that the matching function *f*, from every pair of user profiles in a set $\{P_1, P_2,\ldots, P_n : n \text{ is the number of profiles}\}$ where P_1 is fixed, a set of matching parameters *p* (in our case the weights), and a set of external resources *r* (such as WordNet, see Section 2.4), returns the set of similarities $\{ SIM_{11}, SIM_{12},\ldots, SIM_{1n} \}$ between P_1 and all other profiles in a set $\{P_2,\ldots, P_n\}$:

$$\{SIM_{11}, SIM_{12},\ldots, SIM_{1n}\} = f(P_1, P_2\ldots P_n, p, r) \quad (14)$$

Where *n* indicates the number of profiles in the system. The definition above is suitable for the user who has a profile P_1 and 1 indicates the profile number. The first argument *m* in SIM_{mn} is the profile that is compared to all other profiles. For every other profile, the formula has to be changed accordingly. Having *n* profiles, the matching function calculates *(n-1)* similarity scores for every profile. In total, the outcome is *n*(n-1)* scores.

4.4 Semantic similarity metrics for matching function

In Section 4.3 we defined the matching function that uses few types of matching methods. One of them is the semantic method. It takes advantage of external resource to calculate the similarity. In this thesis, we use *WordNet* (see Section 2.4) as a knowledge resource for calculation of semantic similarity.

Furthermore, as in the case with syntactic method, we were interested in finding the best performing algorithm that would be used in the PCS. Following sections explain our findings and design choices regarding semantic method.

Literature findings

Budanitsky and Hirst (2006) evaluate five metrics that use *WordNet* as their knowledge resource. They selected five following algorithms: *Hirst and St-Onge*, *Jiang and Conrath, Leacock and Chodorow, Lin, and Resnic* (see Appendix H for further details on these algorithms) and evaluated them in two ways.

Firstly, Budanitsky and Hirst (2006) used original studies of Miller and Walter (1991) and Rubenstein et al. (2000). Miller and Walter used 65 pairs of words which varied from "highly synonymous" to "semantically unrelated" and asked 51 human to rate them on the scale from 0

to 4 according to their “similarity of meaning”. Similarly, Rubenstein et al. took 30 pairs of words from the original 65 pairs and obtained similarity scores from 38 participants. Budanitsky and Hirst (2006) took the advantage of the results of Miller and Walter (1991), and Rubenstein and Goodenough (1965) and compared them to the relatedness scores produces by five algorithms mentioned earlier.

Secondly, Budanitsky and Hirst (2006) evaluated the performance of algorithms in the framework of a particular application: the detection and correction of real-world spelling errors. Authors took 500 articles, removed proper nouns and stop-list words. Then they purposely placed a spelling error every 200 words in the text. For example, they changed the original word by its variation found in the *WordNet*. Authors used five algorithms of semantic relatedness to find and correct spelling errors.

The more detailed description of these experiments is outside of the scope of this thesis; however, we need to know that the *Jiang and Conrath* metric outperformed other four metrics in both sessions of the experiment by Budanitsky and Hirst (2006).

Conclussion

We take the results of the Budanitsky and Hirst (2006) as the main contribution to our decision choices regarding semantic method. Authors compared the performance of semantic similarity metrics in detecting real-world spelling errors and concluded that a metric *Jiang and Conrath* was found superior to those proposed by other scientists. With respect to this, we implemented metric *Jiang and Conrath* in PCS to calculate semantic relatedness between values in user profiles.

4.5 The User Profile/Patient Information Model

According to Lee (1999), the information model is a sharable, stable, and organized structure that represents the concepts, relationships, constraints, rules, and operations. It specifies the data semantics for a specific domain of use. Moreover, there are few approaches for information model modeling such as entity-relationship approach, the functional modeling approach, and the object-oriented approach (Lee, 1999). The aim was to design the information model at the higher level of data requirements and not on functional capabilities of the system.

The Design

There are few possible design choices concerning the entities in the information model. Choosing an appropriate design is a judgment that must be made at the beginning of the modeling process. In the healthcare, every individual can play various roles. As an example, there is possibility of playing few roles at the same time, e.g. an individual can be a patient and a caregiver at the same time. In this study, however, there is no distinction between the role of a *patient* and the *individual* because every individual is considered to play a role of patient. Moreover, having more roles introduces various complexities in the design of the information

model: various hierarchies are possible, different roles, and different parties (see Appendix B for an alternative information model). To use such advanced model there is a need for high-level matching strategy which describes how various entities should be compared. This study however, focuses on matching problem on property-level, e.g. the comparing process does not take into account the possibilities of having other parties than an *Individual* and having other party roles that a *Patient*.

In conclusion, the information model was designed focusing on two major dimensions: personal information about a patient, and preferences for group exercise sessions. In this respect, the model was designed for CVD patients and reflects the properties of group exercise session. With respect to this, the functionality of the PCS is based on the assumption that every user of the system, who has a user profile, is a CVD patient which is interested in participating in group exercise sessions with other patients.

The information model fulfills three tasks (Kobsa, 1993):

- *User subgroup identification*: we have to identify the part of population which is considered to be possible user for the application and the user model itself. We focus on service for group exercise sessions and, particularly, patients with CVD are which have an interest in participating in group exercise sessions.
- *Identification of key characteristics*: there is a need to identify a number of key characteristics that characterize the user. In this study, the user is defined as a CVD patient, therefore the preferences for group exercise sessions and personal information are the factors which distinguish the patients the most.
- *Representation in (hierarchically ordered) entities*: in case of analyzing more than one role of system user, e.g. patient, doctor, and various organizations, there has to be a formal way to represent all the entities in the model. If one of the entities has the subset of properties of another entity, then the relationship can be described in hierarchies, thus the most general characteristics of the entities should be contained in the top most entity. This task is not fulfilled as the matching scenario includes only one role.

Moreover, according to Amato and Straccia (1999) there are two dimensions on which the information model could focus: *what* information has to be represented, and *how* this information has to be represented. Our patient information model is focused on the *what* dimension because the matching function calculates similarity between properties of the entities in the information model.

Furthermore the goal was to design the model that represents two major aspects. These two aspects are the canonical user model and the exercise-specific user modeling (see Section 2.2). The canonical model represents the general properties of users and the exercise-specific model aims to represent specific properties of the group exercise sessions. These aspects are defined as two separate information classes: “Personal information” and “Group exercise preference”

(see Figure 13). By having two classes, we logically distinguished the “classical” human factors from the “specific” health-related factors.

Personal information sub-model: this sub-model represents the user. We investigated three information models, namely: (*i*) SID (Information Framework (SID)), (*ii*) OpenSocial (OpenSocial Project), and (*iii*) the model which was proposed by Golemati et al. (2007) (see Section 2.1). All three models come from different backgrounds: the first is more market-driven (SID model), the second is popular among various social networking application (OpenSocial model), and the last is popular in the scientific literature. We looked at each of them separately and in combination to find the union of the properties used in more than one model.

The entity in the information model for this sub-model is called “Individual”.

Group exercise preference information sub-model: Casillas et al. (2007) give three dimensions that could describe physical group exercise sessions, namely: (*i*) *intensity,* (*ii*) *frequency*, and (*iii*) *duration*. In addition to this, we propose two additional properties which characterize the group exercise session: exercise type and preferred language. The final information model can be found in Figure 13 and the class is called “Group exercise preference”.

For the model to be complete in terms of assumptions described above, we introduced one more entity “Language”. This entity contributes to “Individual” and “Group exercise preference” by allowing multiple relationships. The final table representing our patient information model can be found in the Appendix A and the entity relationship diagram can be found in the Figure 9.

Data types in the information model

There are three types of data that we use in the patient information model: *free text*, *number*, and *a list of options*. We needed three types because various properties in the user information model can be represented in different ways. The Table 11 shows the design choices that we made about data types in the information model.

In order to meet the requirements of matching function, we made a few restrictions for data types listed earlier:

- *Free text* properties are allowed to be maximum of one word except properties “Family Name” and “Given Name” which do not participate in the matching function. Furthermore, only nouns are allowed because semantic matching requires the input to be in the form of nouns.
- *Number* in the user profile must be positive or equal to zero. There are no restrictions for maximum number except computer-specific restrictions.
- *List of options* is a list of items of which the user can choose one. There are different lists of options for various properties such as language, or disabilities.

Figure 13. The entity relationship diagram

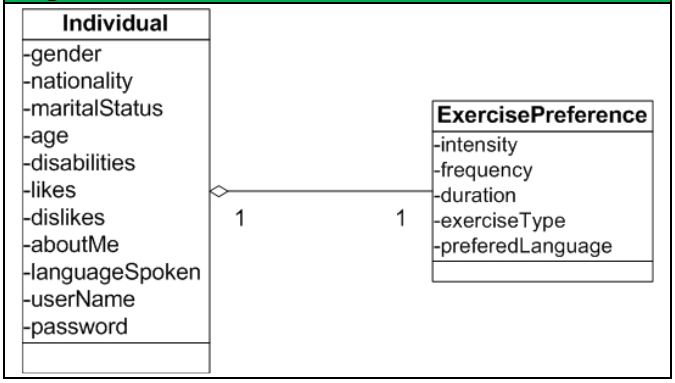


Table 11. The data types of properties

Free text	Number	List of options
Likes, dislikes, given name, family name, about me.	Age, intensity, frequency, duration, exercise type.	Gender, nationality, marital status, country, language id, preferred language, disabilities.

4.6 Chapter summary

In this chapter we presented the design of Patient Comparison System which implemented the patient information model. Also, this system was used during the experiments to find weights for properties in the information model.

PCS has the functionality to create, save and modify the user profile. It also lets users to compare two profiles and assign the similarity to a particular pair. PCS was also used to calculate similarities among profiles by using few sets of weights. The statistical data from PCS was used to calculate the weights for every particular user (see next chapter for more detailed description).

5 Results of Evaluation of Similarity Metrics and Weights

This chapter summarizes the results of the evaluation that follows the methodology in the Chapter 3. First, we present the results of the application of methodologies to evaluate software packages and syntactic similarity metrics (see Section 3.3). During the experiments we used human judgments to conduct the selection process. Next we present the results of weights selection process that consisted of few steps (see Section 3.3). The results were achieved by using correlation and linear regression methods. Specifically, we calculated the correlation between calculated similarities by PCS and assigned similarities by the participants in the experiment. After that we selected the set of weights that best represents the judgment of participants. Also we calculated the weights by using linear regression method with additional software applications Microsoft Office Excel and MathWorks Matlab.

5.1 Results of Methodology Applications

This section presents the results of the evaluation of the software packages and the results of the syntactic similarity metrics. The methodology and the precise application of it are described in Chapter 3. Here we present the outcome of the application that follows the figure below.

The results of software package evaluation

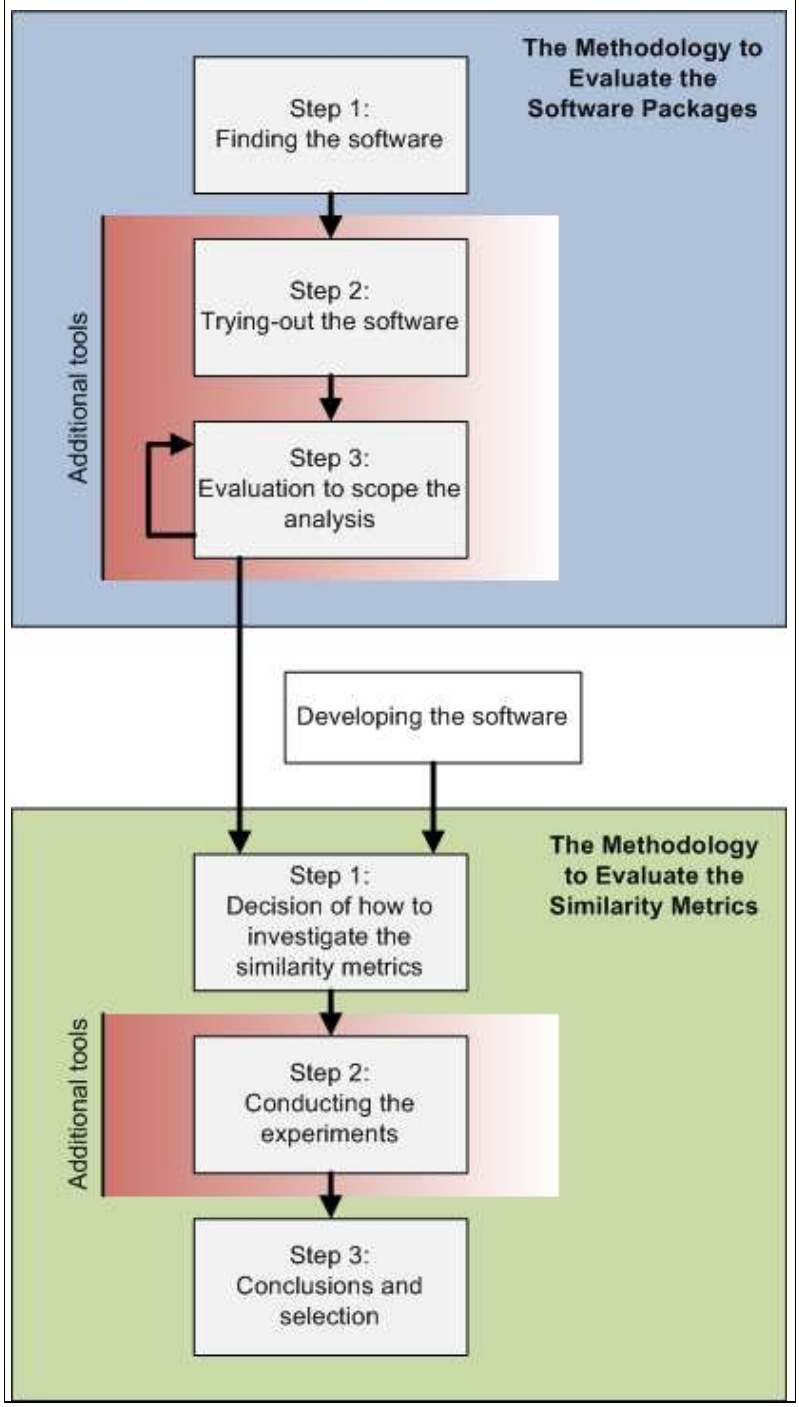
Step 1 (Finding the software): as we mentioned in Section 3.1, we decided to find software packages that implement syntactic similarity metrics instead of developing them. We found four open source software packages on the Internet that implements syntactic metrics: (SecondString Project), (SimPack Project), (SimMetrics Project), and (S-Match Project). The initial task was to select one package for further use.

Step 2 (Trying-out the software): we were able to try out only two software packages out of four. Our lack of prior knowledge in using these packages, poor or no documentation could have led to the fact that only “SimMetrics” and “SecondString” were investigated in the next step. The other two packages did not participate in the later investigation.

Step 3 (Evaluation to scope the analysis): following the methodology in Section 3.1 we listed the similarity metrics that have equal names in “SimMetrics” and “SecondString”. The list of metrics that are implemented in both packages is the following: *Jaro, Jaro Winkler, Jaccard, Monge Elkan, Levenstein, Needleman Wunch,* and *Smith Waterman.* All other metrics either have different names or are implemented only in one of the packages. From the initial analysis we observed that not all the similarity metrics which are implemented in both packages calculate exactly equal results even if they have the same name. The list of similarity metrics which give the same or almost equal results in both packages is the following: *Jaro, Jaro-*

Winkler, Jaccard, and Monge Elkan. Other metrics produced various results such as negative numbers, not normalized numbers or no result at all. Possible programming errors or lack of our knowledge might have been the reasons for such results.

Figure 14. The methodology to evaluate the similarity metrics



Next, we experimented with these metrics as described in Section 3.3. We changed the input strings and monitored the differences among similarity calculations between “Simmetrics” and “SecondString”.

The Appendix K shows the exact differences between similarities when only one word was used as an input. The Table 12 shows the average calculation differences and standard deviations for every similarity metric.

Table 12. Differences among similarity metrics in SimMetrics and SecondString by using one word

	Jaro	Jaro Winkler	Jaccard	Monge Elkan
Standard deviation	7.43E-09	7.43E-09	0	0
Average	8.81E-09	8.81E-09	0	0

The numbers in the table above indicate the difference between two packages. *Jaro* and *Jaro Winkler* metrics have differences in the calculation between packages SimMetrics and SecondString. These differences are small; however, they might be higher if we change the input strings for this experiment.

Next we changed the input strings from being one word to three words. Table 13 shows the standard deviations and average calculation differences in this experiment.

Table 13. Differences among similarity metrics in SimMetrics and SecondString by using three words

	Jaro	Jaro Winkler	Jaccard	Monge Elkan
Standart deviation	1.55E-08	1.54E-08	0	0
Average	1.55E-08	1.57E-08	0	0

The differences were still small but they were higher than in the Table 12.

The next experiment included the input string that was a full sentence. The table below shows the calculation differences among the similarity metrics.

Table 14. Differences among similarity metrics in SimMetrics and SecondString by using one word

	Jaro	Jaro Winkler	Jaccard	Monge Elkan
Standart deviation	1.7E-01	1.7E-01	0.37E-01	0.69E-01

Average	0.73E-01	0.73E-01	0.22E-01	0.41E-01

It can be seen that the calculation differences with a full sentence were highest.

In conclusion, the experiments described earlier indicate differences of packages *SecondString* and *SimMetrics* in terms of calculation results. However, these experiments do not give indications about performance of the similarity metrics in comparison to each other. Moreover, we did not find any patterns in differences among experimental results. Third, some cases during the experiments produced unexpected results which demonstrated high computational differences between the software packages. For example, the case with the input "The geared throughput invokes the nuisance underneath its arranged rocket." and "A potential drip reasons." gave the result of 0.574 which is a high difference knowing that it can vary from 0 to 1 (see the Appendix K for the complete table).

In line with computational differences, we analyzed the source code of the two packages to find out the reasons that have led to these differences.

First, we checked if the metric name and the source code in the package correspond to the mathematical formula defined by original authors of the metric. Second, we paid attention to additional parameters used in the source code. Third, we analyzed how different metrics interpret the input.

It is worth to mentions that there is not much documentation that explains the use various metrics in *SecondString* and in *SimMetrics*.

Despite the distinctive programming style between the packages we found a few fundamental differences in source code.

First, we found an error in the source code which resulted in different similarity scores. In detail, *SecondString* implements *Jaro* and *Jaro Winkler* metrics. *Jaro Winkler* uses *Jaro* metric as part of its formula (see Appendix F). Moreover, *Jaro* metric uses a definition of *common character.* According to the original definition, the common character "must be within **half** the length of the shorter string". Then the definition of the **half** in this context should be as shown below:

$$half = \frac{min(|s|, |t|)}{2} \quad 15)$$

Where $|s|$ and $|t|$ are the lengths of strings *s* and *t*. In the source code of *SecondString* the half is implemented as:

Figure 15. The programming code of the "half" in SecondString

```
private int halfLengthOfShorter(String str1,String str2)
{
     return(str1.length()>str2.length())?str1.length()/2+1:str2.length()
     /2 +1;
}
```

Mathematically this piece of source can be expressed as:

$$half = \frac{max(|s|,|t|)}{2} + 1 \qquad 16)$$

This definition does not follow the original formula by Jaro and this fact leads to the conclusion that, even the metric is called by the name of *Jaro*, it calculates different results comparing to other implementations.

Second, *Monge Elkan* metric in SecondString uses a constant by a name *char_exact_match_score* which is assigned to an integer number of five. This constant is used in the calculation process (see the Figure 15 below).

Figure 16. The programming code of *Monge Elkan* metric in *SecondString*

```
public double score(StringWrapper s,StringWrapper t) {
      if (scaling) {
            int minLen = Math.min( s.unwrap().length(),
t.unwrap().length() );
            return super.score(s,t) / (minLen *
CHAR_EXACT_MATCH_SCORE);
      } else {
            return super.score(s,t);
      }
}
```

Due to the lack of documentation it is not clear what are exact the reasons to use this parameter to calculate the similarity value and why the value is equal to five.

Third, real numbers in both packages are reflected by different primitive data type of Java programming language. In *SecondString* it is float and in *SimMetrics* it is double. The double data type has a twice higher precision than float, thus this difference causes minor differences in results between packages.

To sum up, from the results that are described we concluded that the software package SimMetrics outperformed SecondString. We came to this conclusion by showing that SimMetrics and SecondString are different in terms of calculated similarity values. In some cases these differences are quite substantial. Moreover, during the analysis of source code we found that

Jaro and *Jaro Winkler* metrics are used in a distinctive ways. In SecondString, these algorithms do not follow the definitions of the original metrics. Nevertheless, *Jaro* and *Jaro Winkler* are one of most used algorithms in the field.

We considered these reasons to be meaningful to make a decision to use *SimMetrics* as the software for further analysis of the similarity metrics and the following sections present the results of similarity metrics evaluation that follows the methodology in Section 3.2.

The results of similarity metrics evaluation

Step 1 (Decision how to investigate the similarity metrics): according to the methodology, this phase does not produce any calculation results except the action plan of how to achieve them.

SimMetrics implements 19 similarity metrics. However, we were interested to select smaller number of them. To reach this objective, further analysis of *SimMetrics* source code was conducted. We examined source code of each metric and attempted to narrow the analysis to most suitable metric.

After the analysis of source code we observed tree main issues that helped to narrow the list of similarity metrics:

- Some metrics treat input strings as having more than one word. In this case, if two words are not completely equal, the similarity would be 0. Otherwise 1. Following the assumption that the input string is just one word, the metrics that follows this paradigm were eliminated from the list.
- We also eliminated some metrics that use additional parameters in the process of calculation. Most of the times these parameters have no documentation and reasoning.

In the end we listed only nine similarity metrics that do not have issues described above. These metrics were: *Chapman Mean Length, Levenshtein, Jaro, Jaro Winkler, Smith Waterman Gotoh Windowed Affine, Needleman Wunch, Smith Waterman, Smith Waterman Gotoh, Monge Elkan*.

Step 2: (Conducting the experiments): both sessions of the experiment provided slightly different result that can be seen in the Figure 17 and the Figure 18.

During the first session, three best metrics were *Chapman Meal Length*, *Levenstein*, and *Jaro Winkler*. They scored accordingly: 21, 20, and 16 pushes in *Vesuvius*. Jaro scored 10 pushes, and all other metrics scored less than 10 pushes.

The second session produces equal results in terms of top three metrics. The scores were 20 (*Chapman Meal Length*), 17 (*Levenstein*), and 11(*Jaro Winkler*). However, *Monge Elkan* had 5 pushes during the first session and during the second session it had already 9 pushes which is almost like *Jaro Winkler* (11 pushes).

Combined numbers of both of sessions are shown in Figure 18. The most popular metric was *Chapman Mean Length* that scored 41. The second and third were Levenstein(37) and Jaro Winkler(27). The lowest performance had *Smith Waterman Gotoh Windowed Affine* (13) and *Smith Waterman Gotoh* (13). In total, there were 200 pushes. *Chapman Mean Length* was pushed 20.5% of all pushes, *Levenstein* was pushed 18.5%, and *Jaro Winkler* was pushed 13.5%. In total, these three metrics were pushed 52.5% of all pushes.

Figure 17. The results of the first session to evaluate syntactic metrics

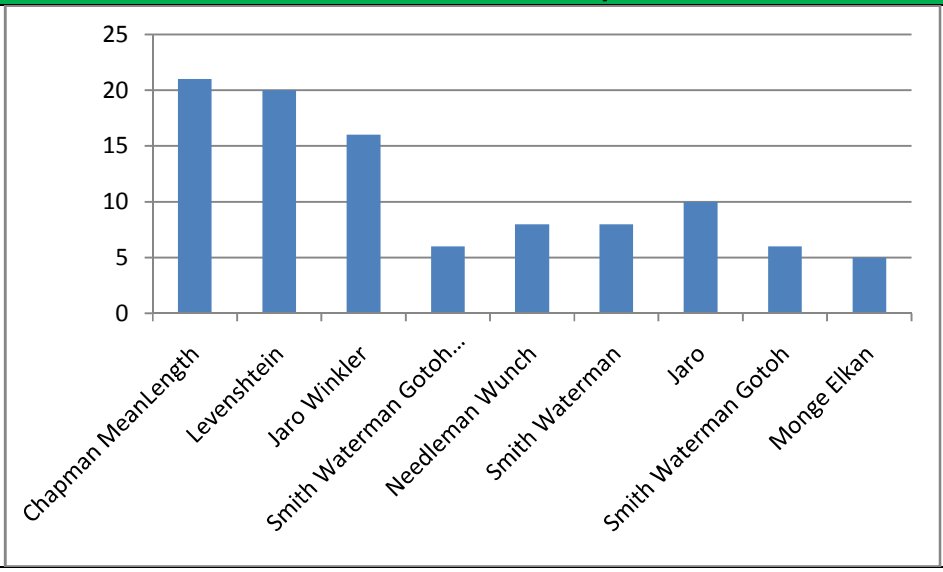


Figure 18. The results of the second session to evaluate syntactic metrics

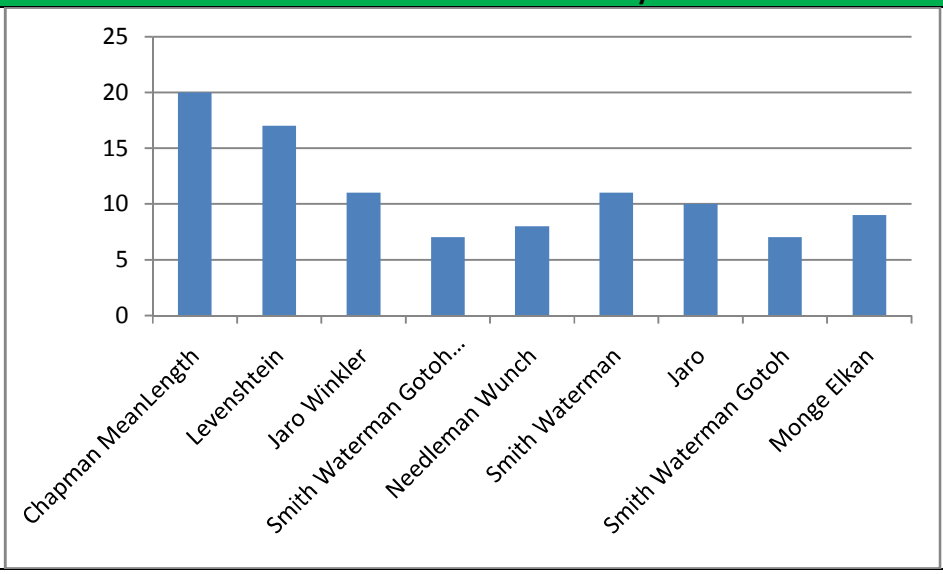
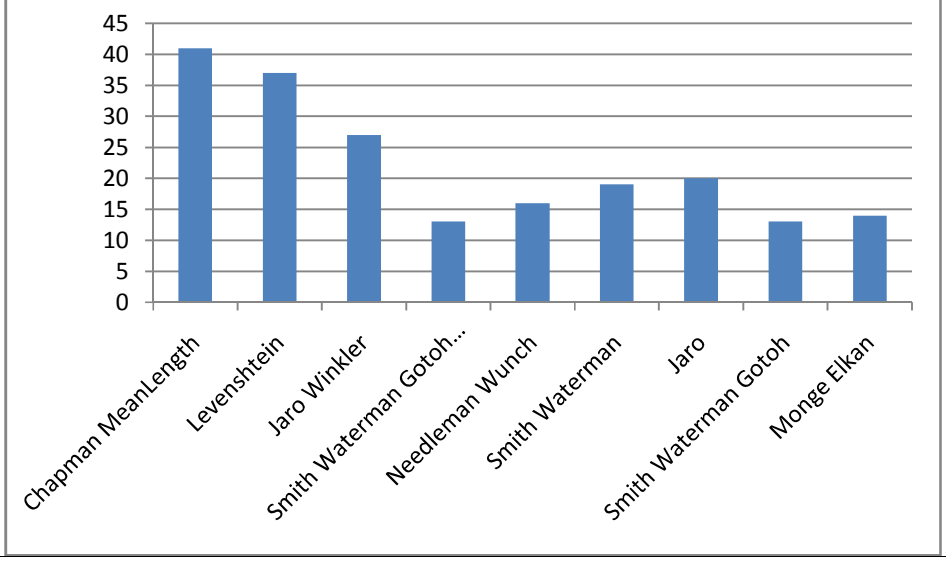


Figure 19. Combines results of both sessions to evaluate syntactic metrics



Step 3 (Conclusions and selection): we observed few imperfections of the proposed experiments. First, during the first session of the experiment people had trouble to think of two nouns as an input for our developed application *Vesuvius*. Some people used to think mostly about the semantically related pair of nouns such as "clock" and "watch". Then, they used to evaluate the similarity based on the perceived semantic relations between such nouns. Second, some metrics calculated equal similarity scores for a pair of words. In such case, a person was in doubt which of the numbers in *Vesuvius* to choose. Third, it was difficult to make participants to evaluate input nouns based on their syntactic similarity and not semantic. People were struggling to understand how the similarity was calculated and how the syntactic similarity metrics work.

From the extensive analysis we made to investigate syntactic similarity metrics in *SimMetrics* we decided that two metrics *Chapman Mean Length* and *Levenstein* showed significant performance during the experiments with *Vesuvius*.

5.2 Results of Selecting Weights of Properties

(Step 1) Results of "Building 8 sets of weights": eight persons were asked to participate in the process of building sets of weights. Each of the participants assigned a weight-number to each of the 13 different properties in the patient information model. The overall table that shows the sets is presented below. The top row in the table indicates the id of the weight and the first column shows the properties that were taken from patient information model (see Section 4.5 for patient information model).

Table 15. The weight sets for patient information model properties								
Weight id / Property	W1	W2	W3	W4	W5	W6	W7	W8
Gender	20	40	10	80	30	15	20	45
Nationality	40	30	10	50	50	55	60	80
Marital status	30	10	30	40	20	70	45	5
Age	50	20	70	10	50	50	30	30
Disabilities	55	40	60	90	60	40	70	70
Likes	60	70	55	80	70	80	40	85
Dislikes	57	90	30	90	75	80	60	50
About Me	70	50	60	50	60	20	30	70
Language spoken	60	10	90	35	10	50	80	20
Intensity	75	70	30	60	55	40	50	80
Duration	50	55	90	45	60	55	80	40
Frequency	35	55	70	70	80	40	65	65
Exercise type	70	60	30	90	80	70	70	70

It is worth to mention that all participants had to use the numbers from 1 to 100.

(Step 2) Results of "Running the matching function among all profiles": after all sets of weights were built, we triggered the matching function in PCS to calculate similarities between profiles by using the weights in the table above. The table that represents all calculated similarities by PCS is shown below. The first column in the table lists all combinations between the eight profiles in PCS.

Table 16. Calculated similarities between profile pairs								
Weight id / Profile pair	W1	W2	W3	W4	W5	W6	W7	W8
P1, P2	0.276	0.312	0.283	0.289	0.272	0.22	0.262	0.296
P1, P3	0.378	0.338	0.533	0.359	0.373	0.402	0.413	0.316
P1, P4	0.463	0.487	0.545	0.534	0.527	0.431	0.507	0.493
P1, P5	0.264	0.229	0.44	0.19	0.277	0.268	0.311	0.213
P1, P6	0.385	0.353	0.42	0.321	0.363	0.323	0.363	0.358
P1, P7	0.368	0.38	0.459	0.398	0.404	0.363	0.421	0.358
P1, P8	0.191	0.242	0.295	0.246	0.232	0.192	0.247	0.211
P2, P3	0.306	0.329	0.328	0.39	0.357	0.292	0.338	0.339
P2, P4	0.285	0.253	0.392	0.296	0.27	0.342	0.329	0.224
P2, P5	0.205	0.196	0.317	0.204	0.232	0.196	0.256	0.203

P2, P6	0.446	0.439	0.483	0.378	0.435	0.401	0.471	0.402
P2, P7	0.255	0.235	0.343	0.301	0.271	0.303	0.292	0.236
P2, P8	0.236	0.252	0.335	0.29	0.283	0.224	0.265	0.266
P3, P4	0.212	0.242	0.319	0.244	0.249	0.209	0.247	0.222
P3, P5	0.361	0.327	0.462	0.254	0.339	0.312	0.362	0.314
P3, P6	0.312	0.266	0.357	0.28	0.336	0.304	0.318	0.284
P3, P7	0.426	0.442	0.528	0.424	0.423	0.359	0.46	0.427
P3, P8	0.41	0.463	0.485	0.467	0.437	0.342	0.458	0.454
P4, P5	0.276	0.268	0.429	0.242	0.308	0.257	0.325	0.258
P4, P6	0.193	0.166	0.284	0.161	0.195	0.186	0.217	0.165
P4, P7	0.433	0.41	0.513	0.475	0.471	0.48	0.484	0.399
P4, P8	0.203	0.231	0.304	0.233	0.235	0.203	0.237	0.209
P5, P6	0.263	0.24	0.363	0.273	0.27	0.32	0.291	0.217
P5, P7	0.389	0.364	0.49	0.331	0.376	0.324	0.419	0.369
P5, P8	0.403	0.394	0.412	0.351	0.387	0.352	0.42	0.371
P6, P7	0.17	0.14	0.253	0.149	0.179	0.159	0.19	0.156
P6, P8	0.156	0.154	0.234	0.168	0.179	0.149	0.196	0.162
P7, P8	0.387	0.425	0.466	0.398	0.404	0.328	0.412	0.405

(Step 3) Results of "Let the people to assign the similarity score between profiles":

PCS had the functionality to assign a number to every pair of profiles. Every person who submitted a profile in PCS was asked to assign a number to a pair of profiles. Regarding the methodology in Section 3.3 every participant assigned seven numbers to seven different pairs of profiles. For example, a person with profile id P1 evaluated pairs such as (P1, P2), (P1, P3), (P1, P4), (P1, P5), (P1, P6), (P1, P7), (P1, P8).

The process of assignment resulted in similarity matrix between all eight profiles (see Section 4.3 for similarity matrix).

Table 17. The similarity matrix

Id	P1	P2	P3	P4	P5	P6	P7	P8
P1		0.45	0.7	0.6	0.4	0.2	0.45	0.3
P2	0.2		0.5	0.6	0.25	0.5	0.4	0.7
P3	0.55	0.4		0.7	0.6	0.6	0.75	0.55
P4	0.8	0.35	0.4		0.8	0.3	0.75	0.4
P5	0.3	0.3	0.65	0.5		0.7	0.3	0.2
P6	0.4	0.6	0.2	0.5	0.4		0.3	0.25
P7	0.7	0.3	0.1	0.5	0.35	0.2		0.5
P8	0.55	0.4	0.5	0.3	0.25	0.5	0.3	

It is visible in the table above that people tended to assign different values to symmetric pairs of profiles; e.g., SIM(P1, P3) and SIM (P3, P1) are assigned different values of 0.45 and 0.7. For that reason we calculated average similarities of symmetric pairs that are shown in the Table 18.

Table 18. The averaged similarity matrix

id	P1	P2	P3	P4	P5	P6	P7	P8
P1		--	--	--	--	--	--	--
P2	0,325		--	--	--	--	--	--
P3	0,625	0,45		--	--	--	--	--
P4	0,7	0,475	0,55		--	--	--	--
P5	0,35	0,275	0,625	0,65		--	--	--
P6	0,3	0,55	0,4	0,4	0,55		--	--
P7	0,575	0,35	0,425	0,625	0,325	0,25		--
P8	0,425	0,55	0,525	0,35	0,225	0,375	0,4	

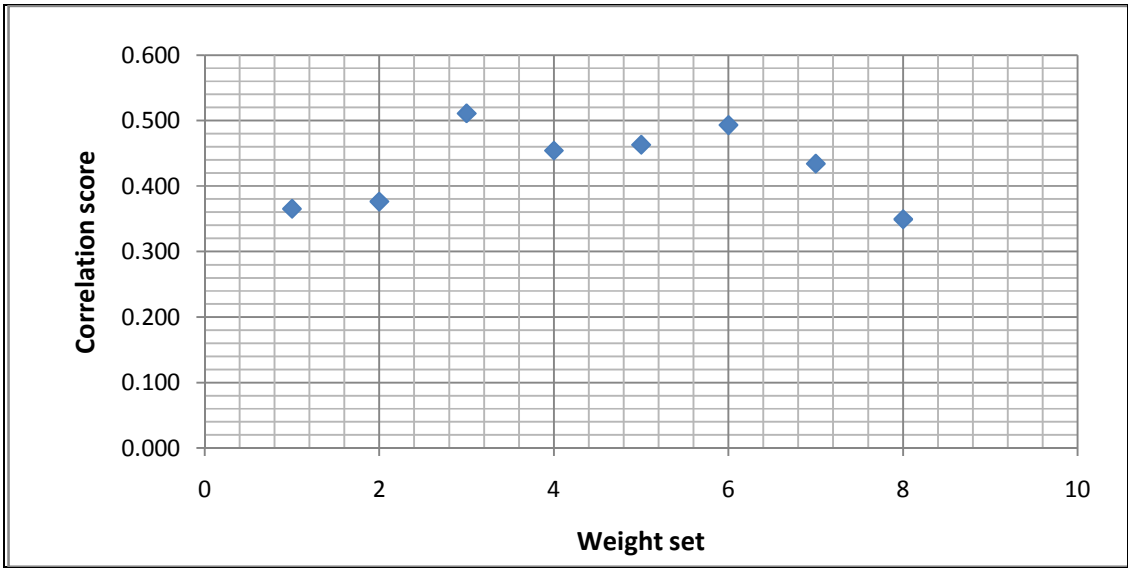
(Step 4) Results of "Compare the similarity values between the matching function and human judgment": having calculated similarities from PCS and the assigned similarities from humans we continued the analysis with the objective to find the best performed set of weights.

We calculated the correlation between the corresponding assigned and calculated similarities. The Table 19 shows the correlation scores (see the Appendix J for the full table).

According to our analysis the highest correlation score has a weight set with the id W6 (0.650). However, weights W1 and W4 scored slightly smaller values of 0.64 and 0.638. The minimum correlation is of the weight set W7 with the score 0.470.

Table 19. The correlation values between assigned and calculated similarities

Weight id	W1	W2	W3	W4	W5	W6	W7	W8
Correlation between human similarities and calculated	0.365	0.376	0.511	0.454	0.463	0.493	0.434	0.349



(Step 5) Results of "Calculate the weights by using linear regression":

The Table 20 shows the similarities among all properties between the profile P1 and all other profiles in PCS. Also the Table 20 shows the assigned similarity by human between these pairs of profiles. We needed this data to be able to calculate the weights for a user with the profile id P1.

Table 20. The similarities among properties between (P1, P2)…(P1,P8) taken from PCS

Profile pair	Gender	Nationality	Marital Status	Age	Disabilities	Likes	Dislikes	About Me	Language Spoken	Intensity	Duration	Frequency	Exercise Type	Assigned similarity
P1, P2	1.00	0.00	0.00	0.40	0.00	0.06	0.07	0.06	0.33	1.00	0.33	0.60	0.00	0.45
P1, P3	1.00	0.00	1.00	1.00	0.00	0.08	0.05	0.52	0.67	0.00	0.83	0.80	0.00	0.70
P1, P4	1.00	0.00	0.00	0.60	1.00	0.25	0.09	0.32	0.33	0.00	0.83	0.90	1.00	0.60
P1, P5	0.00	0.00	0.00	1.00	0.00	0.08	0.06	0.09	0.67	0.00	0.83	0.90	0.00	0.40
P1, P6	0.00	0.00	0.00	1.00	1.00	0.08	0.32	0.07	0.33	1.00	0.33	0.40	0.00	0.20
P1, P7	1.00	0.00	0.00	0.60	0.00	0.08	0.03	0.10	0.67	0.00	0.83	0.90	1.00	0.45
P1, P8	1.00	0.00	0.00	0.20	0.00	0.08	0.06	0.06	0.33	0.00	0.83	0.70	0.00	0.30

In Excel after applying the linear regression method we had a chance to compare the calculated and assigned weights which are presented in Table 21. The Table 21 requires few remarks:

- Some calculated weights are negative which is not allowed in our proposed model.
- Some calculated weights are higher than 100 which is also not allowed.
- Some calculated weights are equal to zero.

Table 21. The calculated and assigned weights for comparison													
Weight id	**Calculated weights by linear regression for every property**												
W1'	14	0	-65	23	-27	0	0	189	0	10	0	0	-2
W2'	5	0	-23	0	-120	0	0	0	0	-73	22	134	-7
W3'	35	0	-46	84	27	0	0	0	0	-4	0	-22	-25
W4'	-11	0	17	-26	-8	0	0	0	0	0	100	0	33
W5'	0	0	41	0	-106	0	0	0	-355	24	-68	360	-9
W6'	-20	0	0	19	1	0	0	0	0	-8	69	13	-19
W7'	-13	0	-31	0	-59	0	0	0	-182	-23	0	219	8
W8'	-4	0	0	-38	-7	0	0	0	0	-16	9	85	-25
	Assign weights by humans for every property												
W1	20	40	30	50	55	60	57	70	60	75	50	35	70
W2	40	30	10	20	40	70	90	50	10	70	55	55	60
W3	10	10	30	70	60	55	30	60	90	30	90	70	30
W4	80	50	40	10	90	80	90	50	35	60	45	70	90
W5	30	50	20	50	60	70	75	60	10	55	60	80	80
W6	15	55	70	50	40	80	80	20	50	40	55	40	70
W7	20	60	45	30	70	40	60	30	80	50	80	65	70
W8	45	80	5	30	70	85	50	70	20	80	40	65	70

Even with some flaws in the table above we calculated the correlation between every corresponding pair of calculated and assigned pair of weights. The Table 22 shows the results.

Table 22. The correlation between calculated and assigned weights	
Weight pair	**Correlation between weight sets**
W1 and W1'	**0.394**
W2 and W2'	**0.067**
W3 and W3'	**0.160**
W4 and W4'	**-0.013**
W5 and W5'	**0.558**

W6 and W6'	**0.098**
W7 and W7'	**-0.117**
W8 and W8'	**0.094**

In the MathWorks Matlab we calculated the weights by using a different function which can apply some restrictions to the calculations. In our case these restrictions were that the weight can be in the range [1; 100]. Nevertheless, Matlab was not able to find a solution to the equation which meant that there is no weight set possible that follows the linear equation in Section 3.4 and has a restriction to calculate weight in the range [1; 100]. In our case the system of equations is underdetermined which means that the number of equations is fewer than the number of variables (properties in our case). Underdetermined systems can be inconsistent which means that they have no solutions.

After such result we tested the same equation but with more statistical data. By doing that we made the system of equation overdetermined by increasing the number of profiles to make it higher that the number of properties in the linear equation. We extended the Table 20 by adding eight more profiles with random similarity values among properties (see the table 23) and again triggered Matlab to calculate the weights. However, Matlab was not able to find such a weight set which would follow the equation in Section 3.4 and the statistical data in the Table 23. We did not do any more modifications to the statistical data or the equation.

Table 23. An extension of the Table 20

Profile pair	Gender	Nationality	Marital Status	Age	Disabilities	Likes	Dislikes	About Me	Language Spoken	Intensity	Duration	Frequency	Exercise Type	Assigned similarity
P1, P2	1	0	0	0.4	0	0.06	0.07	0.06	0.33	1	0.33	0.6	0	0.2
P1, P3	1	0	0	0.4	0.5	0.07	0.05	0.06	0.33	0	0.17	0.8	1	0.5
P1, P4	1	0	1	0.8	0	0.07	0.18	0.06	0.67	0	0.5	0.5	0	0.6
P1, P5	0	0	0	0.4	0.67	0.06	0.06	0.06	0.33	0	0.5	0.7	0	0.25
P1, P6	0	0	0	0.4	0	0.06	0.11	0.09	0.67	1	1	0.8	1	0.5
P1, P7	1	0	1	0.8	0.33	0.05	0.04	0.06	0.33	0	0.17	0.7	0	0.4
P1, P8	1	0	0	0.8	0.5	0.07	0.06	0.05	0.33	0	0.17	0.9	0	0.7
P1, P9	1	0	1	0.5	0	0.4	0.33	0.04	0.33	1	0.33	0.3	0	0.2
P1, P10	0	0	1	0.4	1	0.4	0.23	0	0.67	1	0.33	0.1	0	0.4
P1, P11	0	0	1	0.4	0	0.05	0.12	0	0.33	0	1	0.3	0	0.03
P1, P12	0	0	0	0.8	0.33	0.05	0.05	0.05	0.33	1	0.17	0.8	1	0.6
P1, P13	0	0	0	0.5	0.33	0.02	0.01	0	0.33	0	0.33	0.7	0	0.7
P1, P14	0	0	1	0.5	0	0.3	0.05	0.07	0.67	1	0.5	0.5	1	0.1

P1, P15	0	0	0	0.3	0	0.02	0.08	0.03	0	0	0	0.2	1	0.45
P1, P16	0	0	1	0.3	0.5	0.06	0.3	0.08	0.33	0	1	0.5	0	0.33

5.3 Chapter Summary

Regarding the investigation of similarity metrics in SimMetrics, the results showed that metrics Chapman Mean Length and Levenstein outperformed the other metrics. Moreover, it is worth to mention that during the evaluation we applied few restrictions such as comparing only one word.

Regarding the investigation of the weights, the highest correlation between calculated and assigned similarities was achieved with weight sets W3 (0.511) and W6 (0.493). Moreover, all correlations were between 0.349 and 0.511 which, we think, is a narrow range. Nevertheless, all the correlations are quite similar they might have a significant meaning regarding the best weight for the required matching among profiles. For example, two pairs of profiles can be matched and have equal calculated similarities but people might think that these pairs of profiles are dissimilar when compared. Moreover, W3 and W6 show a negative correlation if we compare the relationship among all weights sets. This means that different properties have highest values in W3 and in W6. While in W3 “Age”, “Language spoken”, “Duration”, and “Frequency” are the most important, in W6 “Marital status”, “Likes” and “Dislikes” are the most important properties. Such differences leads to a discussion that all profiles in the information model can be categorized into few groups. In that case, a user could choose the group to which his/her profile is assigned and the comparison process would take it into account to better match the profiles.

From the linear regression analysis results in Section 5.2 we can conclude that it is difficult to predict the result of weight calculation. In our case, the system of equations was underdetermined which implies that the use software to calculate the weights is limited. The correlation values between the assigned and calculated weights vary from -0.117 to 0.558 in the range [-1, 1]. That shows that for some of the participants PCS calculated more meaningful results and for some even appositively unmeaningful. Moreover, neither Microsoft Office Excel neither MathWorks Matlab was able to calculate the weights that fit the statistical data taken from PCS.

The key observations in this chapter are:

- We tried-out two software packages that calculate similarity based on the syntactic of the input strings.
- Four metrics Jaro, Jaro Winkler, Jaccard, and Monge Elkan are implemented in software packages “SeconString” and “Simmetrics” but calculate different similarities.
- We have chosen “Simmetrics” software package to be the object for further analysis.

- In “Simmetrics” only nine similarity metrics were investigated due to the special use cases and restrictions in PCS.
- For the investigation of weight we sampled eight people to select a set of weights.
- With assigned eight sets of weights we calculated the similarities among all profiles in PCS.
- We used the same array of profiles and asked a sample of people to assign similarity numbers to every pair of profiles.
- In the end we calculated the least standard deviation between calculated and assigned similarity values to select the best performing weights.

6 Conclusions and Discussion

The purpose of this thesis was to support the matching process among user profiles in health care virtual communities. We elaborated in three different aspects. First, we designed and applied a methodology to evaluate software packages and similarity metrics. Second, we designed a patient information model for CVD patients. This model was used in the *Patient Comparison System*. Third, we investigated the weights for properties in the patient information model.

Regarding the first research question, we defined the group exercise session as an event for a group of patients who have main similar preferences for the type of an exercise, intensity, frequency and duration. Regarding the second research question, we defined the patient information model as a set of patient characteristics and a set of preferences for group exercise session. Following the third research question, we characterize the syntactic similarity metric as an edit function between two strings. We adopt the definition of normalized syntactic similarity metric. The first three research questions were answered by doing extensive literature review.

To answer the fourth research question we developed and applied the methodology to evaluate similarity metrics and software packages that implement them. Main criteria of our analysis to find the best performing similarity metric was human opinion about the calculated similarity values between few samples of words. After conducting the experiments and analyzing the statistical data, two metrics *Chapman Mean Length* and *Levenstein* showed significant performance in comparison to other metrics.

Based on our results and with regards to the fifth research question we saw that calculated and assigned similarities have a correlation on the range [0.349; 0.511]. To improve this correlation three aspects can be taken into account. During the assignment of weights and assignment of similarities participants may be more accurate; however we do not conclude that this was an issue. Looking from the PCS perspective the matching function can be changed to calculate similarities more meaningfully for the participants. For example, there are other sources such as WordNet that help to calculate the semantic similarity.

Moreover, we used Microsoft Office Excel and MathWorks Matlab to calculate the weights for participants of the experiment. These two software applications have different functions to calculate linear equations. We wanted to investigate the possibility to calculate the weights for every individual separately by having statistical data and compare these weights to the assigned weights by humans. The results showed that it is complicated to calculate weights based on the statistical data. In our case neither Excel neither Matlab was able to calculate weights that fitted in the required range [1; 100].

Looking back at Chapters 3, 4, and 5, the implications of this thesis are the following. First, the designed patient information model is a starting point to design a comprehensive information model for the patients with CVD. Second, the methodology to evaluate the similarity metrics can

help scientists and practitioners to take advantage of the broad range of similarity metrics. We believe that our systematic approach of evaluation would succeed to make the other research faster and more efficient. Third, we conclude that it is important to get deeper knowledge of the statistical data in order to select one or more best performing sets of weights.

While working on this thesis we used few additional software applications. To be able to match values semantically, we used WordNet database. To apply the methodology to evaluate syntactic similarity metrics we developed two software applications, namely, *Nile* and *Vesuvius*. To implement patient information model and investigate weights we developed the Patient Comparison System (PCS). Finally, similarity metrics in the software package SimMetrics were analyzed in more depth.

PCS was developed to implement patient information model, to calculate similarities among profiles and to help during the analysis of weights. With no more major modifications PCS could be used for further experimentation by changing the patient information model. For example, in this thesis we considered that patients are allowed only one preferred language; however the possibility to choose more than one language could increase the number of good matches among profiles. With regards to the methodology, PCS can be used to calculate the similarities among more than eight profiles. However, by increasing the number of participants, the number of calculations increases extremely rapidly and therefore the calculation time. In that case PCS would calculate the similarities among one hundred profiles for few minutes.

With regards to the methodology, few changes can be made. First, people who participated in the experiments were not patients. For real patients it would be easier to understand medical terminology which is used in patient information model. Secondly, with regards to the evaluation of software packages and investigation of similarity metrics, the technical aspects were not analyzed. For example, it would be meaningful to investigate the similarity metrics from the software efficiency point of view as with more user profiles the cost of calculation increase rapidly.

References

About MathWorks Matlab function LINPROG, from http://www.mathworks.com/help/toolbox/optim/ug/linprog.html

About MathWorks Matlab, from: http://www.mathworks.com/products/matlab/

About Microsoft Office Excel function LINEST, from http://office.microsoft.com/en-us/excel-help/linest-HP005209155.aspx

About Microsoft Office Excel, from: http://office.microsoft.com/en-gb/excel/

About WordNet, Princeton University, from http://wordnet.princeton.edu/

Alignment API, from: http://alignapi.gforge.inria.fr/

Amato, G., Straccia, U. (1999). User profile modeling and applications to digital libraries. In Lecture Notes in Computer Science: vol. 1696. Proceedings of the 3rd European conference on research and advanced technology for digital libraries (ECDL-99), Paris, France (pp. 184–197). Springer-Verlag.

Apache Tomcat, from: http://tomcat.apache.org/

Banerjee, S., Pedersen, T. (2002). An adapted Lesk algorithm for word sense disambiguation using Word- Net. In *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics*, pages 136–145, Mexico City, February.

Brozovsky, L., Petricek, V. (2007). Recommender system for online dating service, in: Proceedings of Znalosti 2007 Conference, VSB, Ostrava.

Budanitsky, A., Hirst, G. (2006). Evaluating wordnet-based measures of lexical semantic relatedness. Computational Linguistics, 32(1):13–47.

Cardiovascular Diseases, World Health Organization, from: http://www.who.int/mediacentre/factsheets/fs317/en/index.html

Casillas, J. M., Gremeaux, V., Damak, S., Feki, A., Perennou, D. (2007). *Exercise training for patients with cardiovascular disease.* Annales de readaptation et de medicine physique (50) p. 403-418.

Cohen, W., Ravikumar, P., Fienberg, S. (2003). A comparison of string distance metrics for name-matching tasks. In *Proc. IJCAI-03 Workshop on Information Integration on the Web*.

Corra, U., Giannuzzi, P., Adamopoulos, S., Bjornstad, H., Bjarnason-Weherns, B., Cohen-Solal, A., et al. (2005). *Executive summary of the position paper of the working group on cardiac rehabilitation and exercise physiology of the European Society of Cardiology (ESC): core*

components of cardiac rehabilitation in chronic heart failure. Eur J Cardiovasc Prev Rehabil; 12:321–5.

Demiris, G. (2006). The diffusion of virtual communities in health care: concepts and challenges. Patient Educ Couns 2006; 62: 178-88.

Distance Between Two Points, from: http://qwickstep.com/search/distance-between-two-points.html

Distance Function, from: http://www.comp.nus.edu.sg/~stevenha/viz/appendixD_distancefunction.pdf

Ehrig, E., Sure, Y. (2004). Ontology mapping - an integrated approach. In Proceedings of the European Semantic Web Symposium (ESWS), pages 76–91.

European Union Language Statistics, Eurostat, from: http://epp.eurostat.ec.europa.eu/cache/ITY_PUBLIC/3-24092009-AP/EN/3-24092009-AP-EN.PDF

Euzenat, J., & Shvaiko, P. (2007). *Ontology Matching* . New York: Springer-Verlag Berlin Heidelberg. G. Salton. *Automatic Information Organization and Retrieval*

Friend of a Friend Project, from: http://www.foaf-project.org/

Golemati M., Katifori A., Vassilakis C., Lepouras G., Halatsis C., (2007). Creating an Ontology for the User Profile: Method and Applications. In *Proceedings of the First International Conference on Research Challenges in Information Science (RCIS)*.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. Knowledge Acquisition 5(2): 199-220

Health Level 7 International, from: http://www.hl7.org/about/index.cfm

Hibernate, JBoss Community, from: http://www.hibernate.org/

Information Framework (SID), TM Forum, from: http://www.tmforum.org/InformationFramework/1684/home.html

Internet Usage Statistics, from: http://www.internetworldstats.com/stats.htm

Java Enterprise Edition, Oracle, from: http://www.oracle.com/technetwork/java/javaee/overview/index.html

Java WordNet Similarity Library, Grid Computing Lab, from : http://grid.deis.unical.it/similarity/index.html?page=main.html&section=download

JavaServer Pages Technology, Oracle, from: http://java.sun.com/products/jsp/

Kang, J. Naughton (2003). *On schema matching with opaque column names and data values.* In: Proceedings of the 22nd International Conference on Management of Data (SIGMOD), pages 205–216, San Diego (CA US), 2003.

Kim, W.G., Lee, C., Hiemstra, S.J. (2004), Effects of an online virtual community on customer loyalty and travel product purchases, Tourism Management, Vol. 25 No. 2, pp. 343-55.

Kobsa, A.(1993). User modeling: Recent work, prospects and hazards. In *Adaptive User Interfaces: Principles and Practice*, T. K. M. Schneider-Hufschmidt and U. Malinowski, Eds. North-Holland, Amsterdam, The Netherlands, 111–128.

L. Z. Rubenstein, K. R. Josephson, P. R. Trueblood, S. Loy, J. O. Harker, F. M. Pietruszka, A. S. Robbins (2000). *Effects of a Group Exercise Program on Strength, Mobility, and Falls Among Fall-Prone Elderly Men*

Lee, Y. T. (1999). Information Modeling: From Design To Implementation. Proceedings of the Second World Manufacturing Congress, ed. S. Nahavandi and M. Saadat, 315-321. Canada/Switzerland. International Computer Science Conventions.

Middleton, S.E., Shadbolt, N.R., de Roure, D.C. (2004). Ontological User Profiling in Recommender Systems, ACM Trans. Information Systems, vol. 22, no. 1, pp. 54-88, 2004.

Miller, G. A., Walter G.Ch. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28.

Miller, G.A. (1995). WordNet: A Lexical Database for English. *Communications of ACM*(11): 39-41.

Mutrie, N., Campbell, A.,M., Whyte, F., et al.(2007). Benefits of supervised group exercise programme for women being treated for early stage breast cancer: pragmatic randomised controlled trial. *BMJ* 2007; 334:517.

MySQL, from: http://www.mysql.com/

Navigli, R. (2009). Word Sense Disambiguation: a survey. *ACM Computing Surveys*, 41(2):1–69, 2009.

OpenSocial Project, from: http://code.google.com/apis/opensocial/

Prosser, G., Carson, P., Phillips, R. (1985). *Exercise after myocardial infarction : Long-term rehabilitation effects. Journal of Psychosomatic Research.*

Random Sentence Generator, from: http://watchout4snakes.com/creativitytools/RandomSentence/RandomSentence.aspx

Random String Generator, from: http://www.random.org/strings/

Razmerita, L., Angehrn, A., Maedche, A. (2003). Ontology based user modeling for Knowledge Management Systems, Proceedings of the User Modeling Conference, Pittsburgh, USA, Springer Verlag, pp. 213-217.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of IJCAI-95*, pages 448–453, Montreal, Canada.

Rich, E. (1983). *Users are individuals: individualizing user models. In. Int. Journal of Man-Maschine Studies, 18, p. 199-214.*

Rubenstein, H., Goodenough, J.B. (1965). Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, October.

Rubenstein, L.Z., Josephson, K.R., Trueblood, P.R., et al.(2000). Effects of a group exercise program on strength, mobility, and falls among fall-prone elderly men. J Gerontol A Biol Sci Med Sci 2000; 55A: M317–21.

Salton, G., McGill, M.J. (1984). *Introduction to Modern Information Retrieval*. McGraw-Hill.

SecondString Project, from: http://secondstring.sourceforge.net/

Shardanand, U., Maes, P. (1995). Social information filtering: Algorithms for automating "word of mouth", in Proceedings of ACM Conference on Human Factors and Computing Systems, pp. 210–217, Association of Computing Machinery, New York.

Sieg, A., Mobasher, B., and Burke, R. (2007). Learning ontology-based user profiles: A semantic approach to personalized. IEEE Intelligent Informatics Bulletin, 8(1):7–17.

SimMetrics Project, from: http://www.dcs.shef.ac.uk/~sam/stringmetrics.html

SimPack Project, from: http://www.ifi.uzh.ch/ddis/simpack.html

S-Match Project, from: http://s-match.org/documentation.html

Sojka, P. (1995) Notes on Compound Word Hyphenation in TEX". *TUGboat* 16(3), 290-297.

Terveen, L. and McDonald, D. W. (2005). Social Matching: A Framework and Research Agenda. ACM Transactions on Computer-Human Interaction, Vol. 12, No. 3, September 2005, Pages 401–434.

Wieringa, R. J. (2003). *Design methods for reactive systems Yourdan, Statemate, and the UML*. Amsterdam [u.a.: Morgan Kaufmann.

Williams, P., Lord, S.R. (1997). Effect of group exercise on cognitive functioning and mood in older women. *Australian and New Zealand Journal of Public Health*, *21*, 45–52.

Winkler, W. (1999). *The state of record linkage and current research problems*. Technical Report, Statistical Research Division, U.S. Census Bureau.

Winkler, W. E. (1990). String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In Proceedings of the Section on Survey Research Methods, American Statistical Association, pages 354-359.

Yang, K. (2010). *Making sense of statistical methods in social research* . Los Angeles, CA [etc.: Sage.

Appendices

A: Patient information model/User profile

Individual data object				
Field Name	**Field Type**	**Description**	**Characteristics, permitted values & units**	**Adopted From**
aboutMe	String	A general statement about the person.	Any text.	OpenSocial, Golemati et al.
age	Number	The age of this Individual.	>0	OpenSocial, Golemati et al.
languagesSpoken	Plural <Language>	List of the languages that the Individual speaks.		OpenSocial, Golemati et al., SID
gender	String	The gender of this Individual.	Male, female.	OpenSocial, Golemati et al., SID
nationality	String	The nationality of this Individual.	Drop down list of country list.	SID
maritalStatus	String	The marital status of this Indivual.	Married, never married, divorced, widowed.	OpenSocial, SID
disabilities	String	The disabilities of this Individual if he/she has some.	Any text.	SID
likes	string	General statement about what the Individual likes regarding the group exercise session.	Any text.	OpenSocial, Golemati et al.
dislikes	string	General statement about what the Individual dislikes regarding the	Any text.	OpenSocial, Golemati et al.

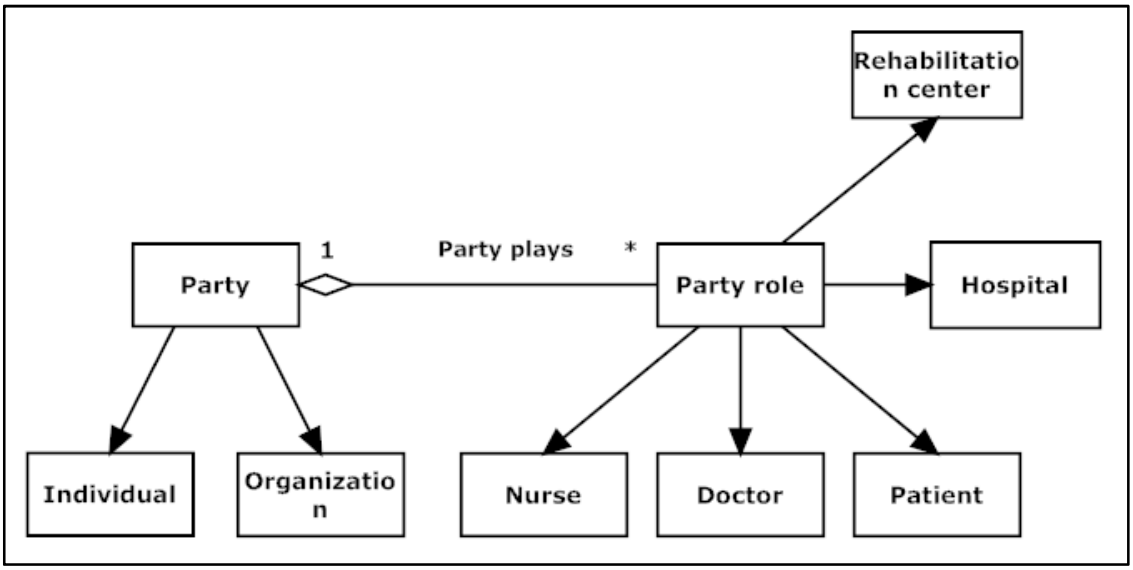
		group exercise session.		
givenName	string	The given name of this Individual, or "First Name" in most Western languages (e.g. Joseph given the full name Mr. Joseph Robert Smarr, Esq.).	Any text.	OpenSocial, Golemati et al., SID
familyName	string	The family name of this Person, or "Last Name" in most Western languages (e.g. Smarr given the full name Mr. Joseph Robert Smarr, Esq.).	Any text.	OpenSocial, Golemati et al., SID
userName	string	An alphanumeric user name, usually chosen by the user, e.g. "jsmarr".	Any text.	
password	string	An alphanumeric user password, usually chosen by the user.	Any text.	

ExercisePreference data object				
Filed Name	**Field Type**	**Description**	**Characteristics, permitted values & units**	**Adopted From**
duration	String	The preferred length of the group exercise session.	Number of minutes.	
intensity	String	The preferred intensity of the group exercise session.	Drop down list.	
frequency	String	The preferred frequency of the group exercise	Number indicating times per week.	

		session during one week.		
exerciseType	String	The preferred type of the group exercise session, e.g. “biking”.	Drop down list.	

B: The alternative information model

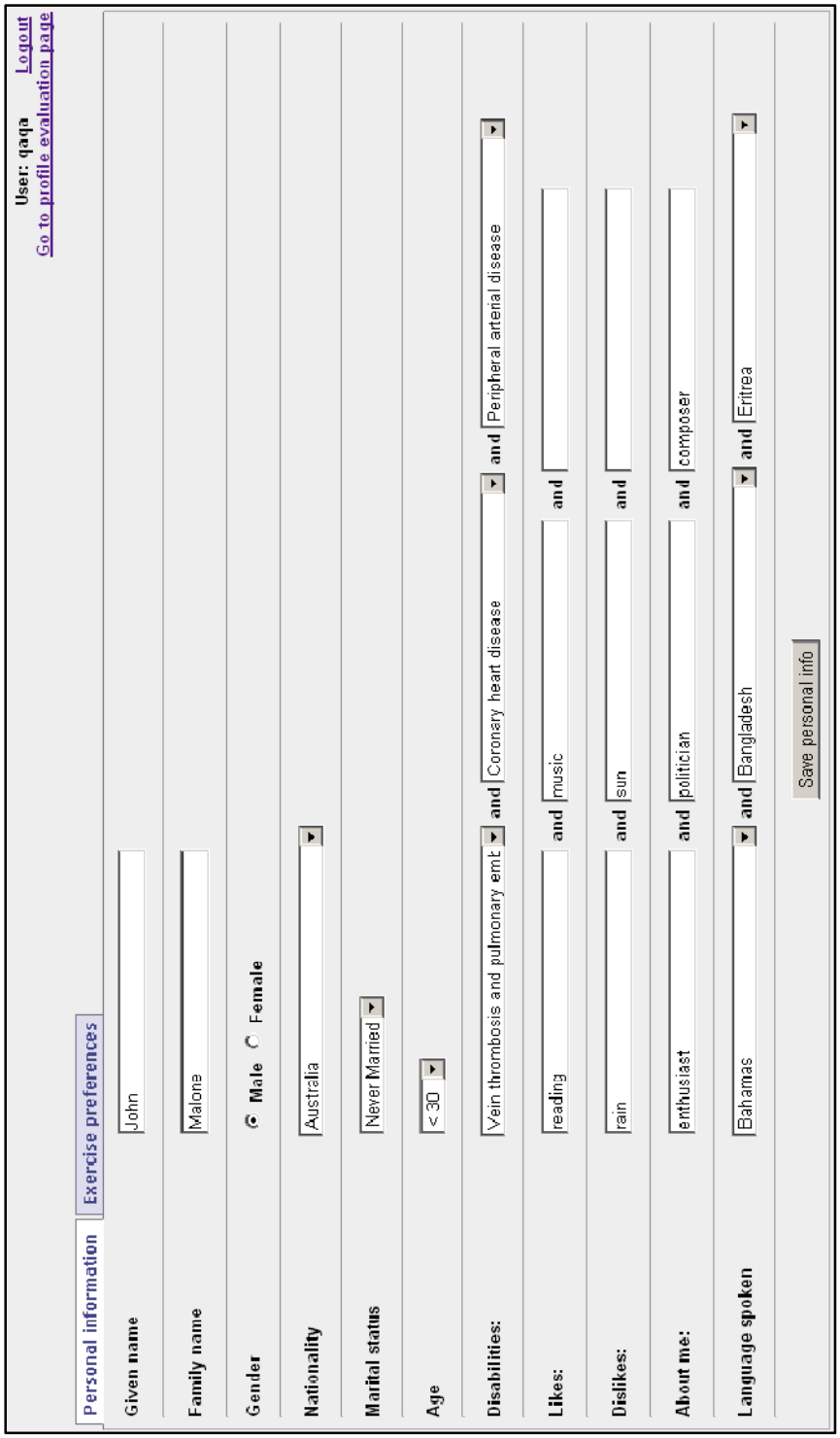
Adopted from SID model (Information Framework (SID)).

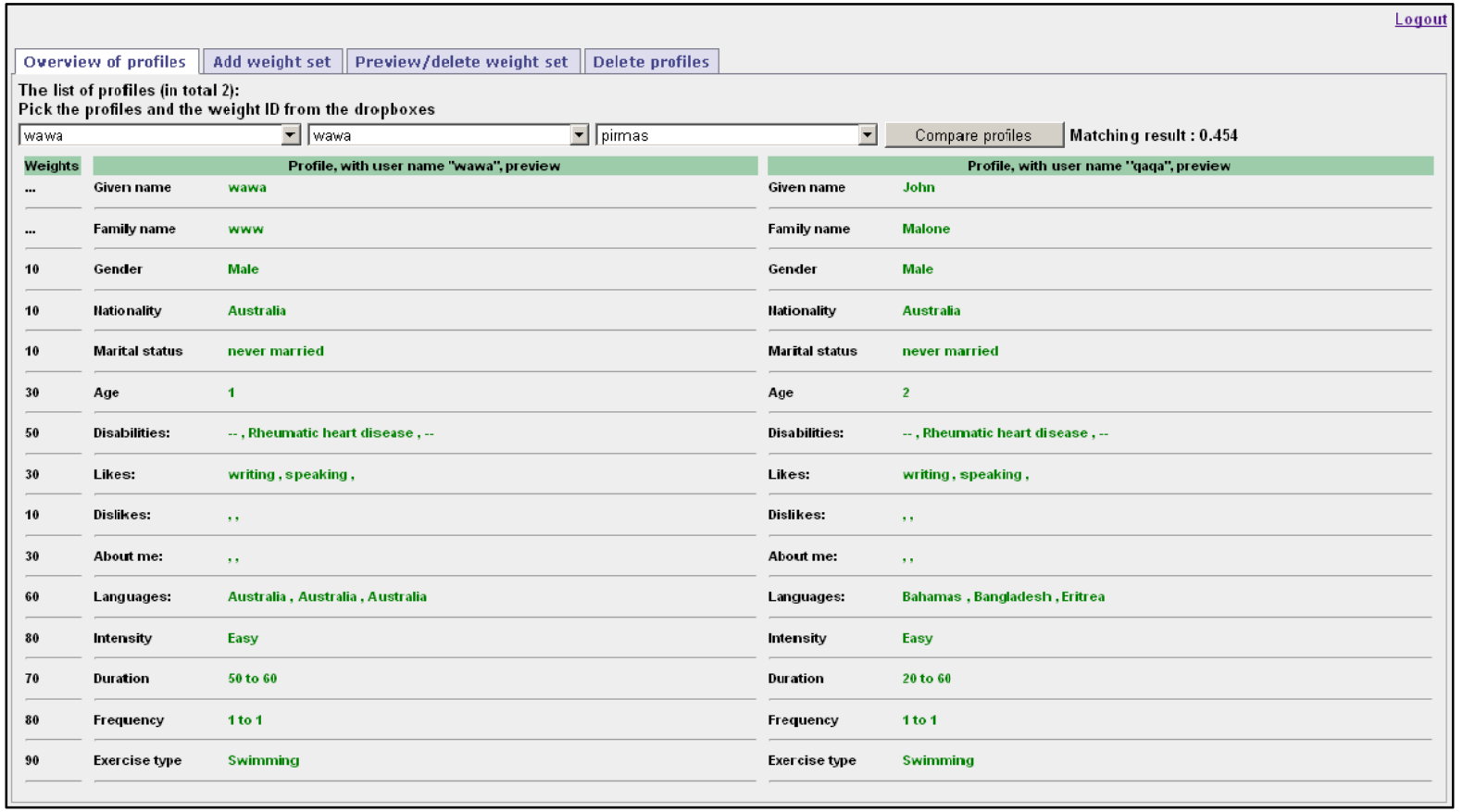


C: The main technologies used for PCS

Name	***Available on***
Apache Tomcat	http://tomcat.apache.org/
JavaServer Pages	http://java.sun.com/products/jsp/
Hibernate	http://www.hibernate.org/
Java EE 5	http://www.oracle.com/technetwork/java/javaee/overview/index.html
MySQL	http://www.mysql.com/about/
Castor	http://www.castor.org/
MyEclipse IDE	http://www.myeclipseide.com/

D: User interface of PCS for patient and caregiver roles





Logout

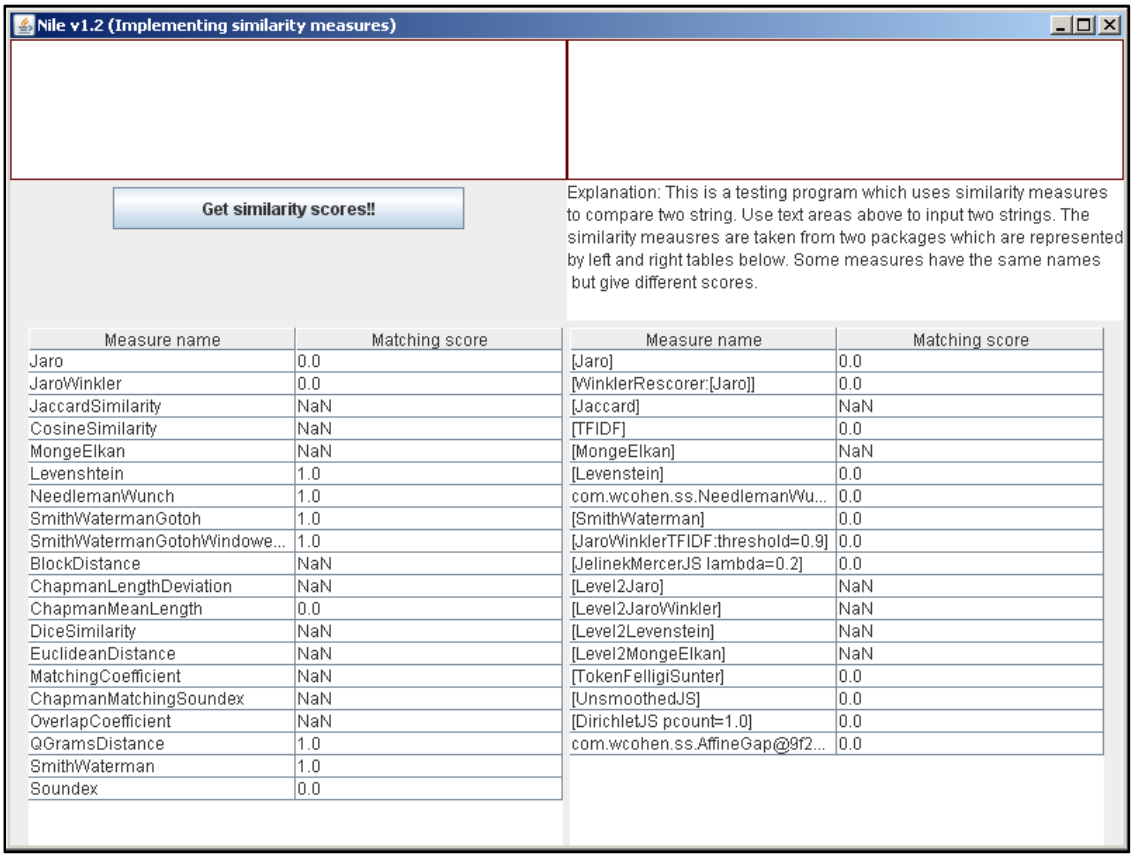
Overview of profiles | Add weight set | Preview/delete weight set | Delete profiles

The list of profiles (in total 2):
Pick the profiles and the weight ID from the dropboxes

wawa | wawa | pirmas | Compare profiles | Matching result : 0.454

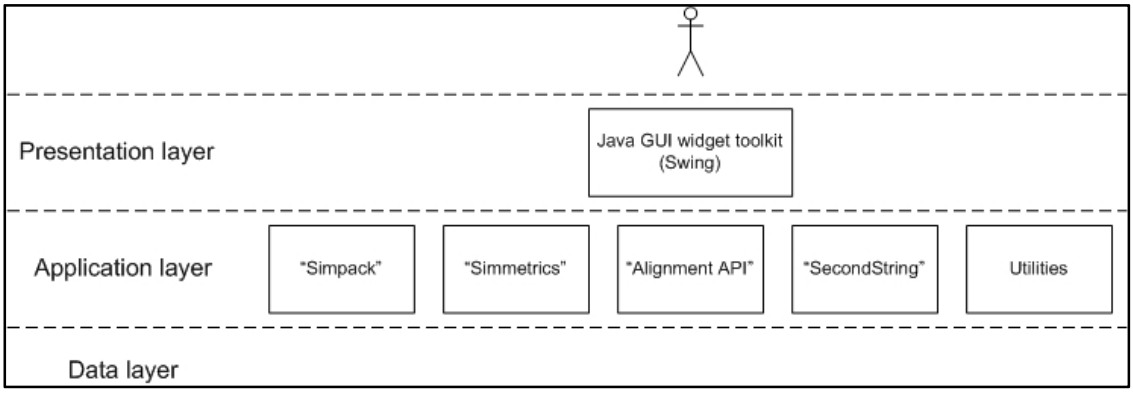
Weights	Profile, with user name "wawa", preview		Profile, with user name "qaqa", preview	
...	Given name	wawa	Given name	John
...	Family name	www	Family name	Malone
10	Gender	Male	Gender	Male
10	Nationality	Australia	Nationality	Australia
10	Marital status	never married	Marital status	never married
30	Age	1	Age	2
50	Disabilities:	-- , Rheumatic heart disease , --	Disabilities:	-- , Rheumatic heart disease , --
30	Likes:	writing , speaking ,	Likes:	writing , speaking ,
10	Dislikes:	, ,	Dislikes:	, ,
30	About me:	, ,	About me:	, ,
60	Languages:	Australia , Australia , Australia	Languages:	Bahamas , Bangladesh , Eritrea
80	Intensity	Easy	Intensity	Easy
70	Duration	50 to 60	Duration	20 to 60
80	Frequency	1 to 1	Frequency	1 to 1
90	Exercise type	Swimming	Exercise type	Swimming

E: Nile application GUI and architecture



Measure name	Matching score
Jaro	0.0
JaroWinkler	0.0
JaccardSimilarity	NaN
CosineSimilarity	NaN
MongeElkan	NaN
Levenshtein	1.0
NeedlemanWunch	1.0
SmithWatermanGotoh	1.0
SmithWatermanGotohWindowe...	1.0
BlockDistance	NaN
ChapmanLengthDeviation	NaN
ChapmanMeanLength	0.0
DiceSimilarity	NaN
EuclideanDistance	NaN
MatchingCoefficient	NaN
ChapmanMatchingSoundex	NaN
OverlapCoefficient	NaN
QGramsDistance	1.0
SmithWaterman	1.0
Soundex	0.0

Measure name	Matching score
[Jaro]	0.0
[WinklerRescorer:[Jaro]]	0.0
[Jaccard]	NaN
[TFIDF]	0.0
[MongeElkan]	NaN
[Levenstein]	0.0
com.wcohen.ss.NeedlemanWu...	0.0
[SmithWaterman]	0.0
[JaroWinklerTFIDF:threshold=0.9]	0.0
[JelinekMercerJS lambda=0.2]	0.0
[Level2Jaro]	NaN
[Level2JaroWinkler]	NaN
[Level2Levenstein]	NaN
[Level2MongeElkan]	NaN
[TokenFelligiSunter]	0.0
[UnsmoothedJS]	0.0
[DirichletJS pcount=1.0]	0.0
com.wcohen.ss.AffineGap@9f2...	0.0



G: Some publications that cite or use the software package SecondString for experiments

- Cohen, W.W., Minkov, E. (2006). A graph-search framework for associating gene identifiers with documents. BMC Bioinformatics, 7(440).
- Tsai, T., Wu, S., Hsu, W. (2005). Exploitation of linguistic features using a CRFbased biomedical named entity recognizer. To appear in ACL Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics, Detroit
- B. Srivastava, B., Mukherjee, D. (2009). Organizing documented processes. In In IEEE Services Computing Conference, Bangalore, India.
- Bronselaer, A., De Tre, G. (2009). A possibilistic approach to string comparison. *IEEE Transactions on Fuzzy Systems*, 17(1):208–223.

I: Entity Person in the OpenSocial framework

Entity Person		
Field Name	**Field Type**	**Description**
aboutMe	string	A general statement about the person.
accounts	Plural-Field <Account>	An online account held by this Person.
activities	Plural-Field <string>	Person's favorite activities.
addresses	Plural-Field <Address>	A physical mailing address for this Person.
age	number	The age of this person. Sometimes sites might want to show age without revealing the specific birthday.
anniversary	Date	The wedding anniversary of this person. The value MUST be a valid Date. The year value MAY be set to 0000 when the year is not available.
appData	Plural-Field <AppData>	A collection of AppData keys and values.
birthday	Date	The birthday of this person. The value MUST be a valid Date. The year value MAY be set to 0000 when the age of the Person is private or the year is not available.
bodyType	string	Person's body characteristics.
books	Plural-Field <string>	Person's favorite books.
cars	Plural-Field <string>	Person's favorite cars.
children	Plural-Field <string>	Description of the person's children.
connected	Boolean	Boolean value indicating whether the user and this Person have established a bi-directionally asserted connection of some kind on the Service Provider's service. The value MUST be either true or false. The value MUST be true if and only if there is at least one value for the relationship field, described below, and is thus

		intended as a summary value indicating that some type of bi-directional relationship exists, for Consumers that aren't interested in the specific nature of that relationship. For traditional address books, in which a user stores information about other contacts without their explicit acknowledgment, or for services in which users choose to "follow" other users without requiring mutual consent, this value will always be false.
drinker	string	Person's drinking status.
displayName	string	Required. The name of this Person, suitable for display to end-users. Each Person returned MUST include a non-empty displayName value. The name SHOULD be the full name of the Person being described if known (e.g. Cassandra Doll or Mrs. Cassandra Lynn Doll, Esq.), but MAY be a username or handle, if that is all that is available (e.g. doll). The value provided SHOULD be the primary textual label by which this Person is normally displayed by the Service Provider when presenting it to end-users.
emails	Plural-Field <string>	E-mail address for this Person. The value SHOULD be canonicalized by the Service Provider, e.g.joseph@plaxo.com instead of joseph@PLAXO.COM.
ethnicity	string	Person's ethnicity.
fashion	string	Person's thoughts on fashion.
food	Plural-Field <string>	Person's favorite food.
gender	string	The gender of this person. Service Providers SHOULD return one of the following Canonical Values, if appropriate:male, female, or undisclosed, and MAY return a different value if it is not covered by one of these Canonical Values.
happiestWhen	string	Describes when the person is happiest.
hasApp	Boolean	Indicating whether the user has application installed.
heroes	Plural-Field <string>	Person's favorite heroes.
humor	string	Person's thoughts on humor.
id	Object-Id	Required. Unique identifier for the Person.
ims	Plural-Field <string>	Instant messaging address for this Person. No official canonicalization rules exist for all instant messaging addresses, but Service Providers SHOULD remove all whitespace and convert the address to lowercase, if this is appropriate for the service this IM address is used for. Instead of the standard Canonical

		Values for type, this field defines the following Canonical Values to represent currently popular IM services: aim, gtalk, icq, xmpp,msn, skype, qq, and yahoo.
interests	Plural-Field <string>	Person's interests, hobbies or passions.
jobInterests	Plural-Field <string>	Person's favorite jobs, or job interests and skills.
languagesSpoken	Plural-Field <string>	List of the languages that the person speaks as ISO 639-1 codes.
livingArrangement	string	Description of the person's living arrangemen.
location	string	
lookingFor	string	Person's statement about who or what they are looking for, or what they are interested in meeting people for.
movies	Plural-Field <string>	Person's favorite movies.
music	Plural-Field <string>	Person's favorite music.
name	Name	The broken-out components and fully formatted version of the person's real name.
networkPresence	Plural-Field <string>	Person's current network status. Specified as one of: AWAY, CHAT, DND, OFFLINE, ONLINE OR XA.
nickname	string	The casual way to address this Person in real life, e.g. "Bob" or "Bobby" instead of "Robert". This field SHOULD NOT be used to represent a user's username (e.g. jsmarr or daveman692); the latter should be represented by the preferredUsername field.
note	string	Notes about this person, with an unspecified meaning or usage (normally notes by the user about this person). This field MAY contain newlines.
organizations	Plural-Field <Organization>	A current or past organizational affiliation of this Person.
pets	Plural-Field <string>	Description of the person's pets
phoneNumbers	Plural-Field <string>	Phone number for this Person. No canonical value is assumed here. In addition to the standard Canonical Values for type, this field also defines the additional Canonical Values mobile, fax, and pager.
photos	Plural-Field <string>	URL of a photo of this person. The value SHOULD be a canonicalized URL, and MUST point to an actual image file (e.g. a GIF, JPEG, or PNG image file) rather than to a web page containing an image. Service Providers MAY return the same image at different sizes, though it is recognized that no standard for describing images of various sizes currently

		exists. Note that this field SHOULD NOT be used to send down arbitrary photos taken by this user, but specifically profile photos of the contact suitable for display when describing the contact.
politicalViews	Plural-Field <string>	Person's political views.
preferredUsername	string	The preferred username of this person on sites that ask for a username (e.g. jsmarr or daveman692). This field may be more useful for describing the owner (i.e. the value when /@me/@self is requested) than the user's person, e.g. Consumers MAY wish to use this value to pre-populate a username for this user when signing up for a new service.
profileSong	string	URL of a person's profile song.
profileVideo	string	URL of a person's profile video.
profileUrl	string	Person's profile URL, specified as a string. This URL must be fully qualified. Relative URLs will not work in gadgets.
published	Date	The date this Person was first added to the user's address book or friends list (i.e. the creation date of this entry). The value MUST be a valid Date.
quotes	Plural-Field <string>	Person's favorite quotes
relationships	Plural-Field <string>	A bi-directionally asserted relationship type that was established between the user and this person by the Service Provider. The value SHOULD conform to one of the XFN relationship values (e.g. kin, friend, contact, etc.) if appropriate, but MAY be an alternative value if needed. Note this field is a parallel set of category labels to the tags field, but relationships MUST have been bi-directionally confirmed, whereas tags are asserted by the user without acknowledgment by this Person. Note that this field consists only of a string value.
relationshipStatus	string	Person's relationship status.
religion	string	Person's relgion or religious views.
romance	string	Person's comments about romance.
status	string	Person's status, headline or shoutout.
scaredOf	string	What the person is scared of.
sexualOrientation	string	Person's sexual orientation.
smoker	string	Person's smoking status.
sports	Plural-Field <string>	Person's favorite sports
tags	Plural-Field	A user-defined category label for this person,

	<string>	e.g. "favorite" or "web20". These values SHOULD be case-insensitive, and there SHOULD NOT be multiple tags provided for a given person that differ only in case. Note that this field consists only of a string value.
thumbnailUrl	string	Person's photo thumbnail URL, specified as a string. This URL must be fully qualified. Relative URLs will not work in gadgets.
turnOffs	Plural-Field <string>	Person's turn offs.
turnOns	Plural-Field <string>	Person's turn ons.
tvShows	Plural-Field <string>	Person's favorite TV shows.
updated	Date	The most recent date the details of this Person were updated (i.e. the modified date of this entry). The value MUST be a valid Date. If this Person has never been modified since its initial creation, the value MUST be the same as the value of published. Note the updatedSince Query Parameter can be used to select only people whose updated value is equal to or more recent than a given Date. This enables Consumers to repeatedly access a user's data and only request newly added or updated contacts since the last access time.
urls	Plural-Field <string>	URL of a web page relating to this Person. The value SHOULD be canonicalized by the Service Provider, e.g.http://josephsmarr.com/about/ instead of JOSEPHSMARR.COM/about/. In addition to the standard Canonical Values for type, this field also defines the additional Canonical Values blog and profile.
utcOffset	Date-UTC-Offset	The offset from UTC of this Person's current time zone, as of the time this response was returned. The value MUST conform to the Date-UTC-Offset. Note that this value MAY change over time due to daylight saving time, and is thus meant to signify only the current value of the user's timezone offset.

J: The assigned and calculated similarities

#	Sim. assigned by / Weight id	W1	W2	W3	W4	W5	W6	W7	W8

	humans									
1	0.325		0.276	0.312	0.283	0.289	0.272	0.22	0.262	0.296
2	0.625		0.378	0.338	0.533	0.359	0.373	0.402	0.413	0.316
3	0.7		0.463	0.487	0.545	0.534	0.527	0.431	0.507	0.493
4	0.35		0.264	0.229	0.44	0.19	0.277	0.268	0.311	0.213
5	0.3		0.385	0.353	0.42	0.321	0.363	0.323	0.363	0.358
6	0.575		0.368	0.38	0.459	0.398	0.404	0.363	0.421	0.358
7	0.425		0.191	0.242	0.295	0.246	0.232	0.192	0.247	0.211
8	0.45		0.306	0.329	0.328	0.39	0.357	0.292	0.338	0.339
9	0.475		0.285	0.253	0.392	0.296	0.27	0.342	0.329	0.224
10	0.275		0.205	0.196	0.317	0.204	0.232	0.196	0.256	0.203
11	0.55		0.446	0.439	0.483	0.378	0.435	0.401	0.471	0.402
12	0.35		0.255	0.235	0.343	0.301	0.271	0.303	0.292	0.236
13	0.55		0.236	0.252	0.335	0.29	0.283	0.224	0.265	0.266
14	0.55		0.212	0.242	0.319	0.244	0.249	0.209	0.247	0.222
15	0.625		0.361	0.327	0.462	0.254	0.339	0.312	0.362	0.314
16	0.4		0.312	0.266	0.357	0.28	0.336	0.304	0.318	0.284
17	0.425		0.426	0.442	0.528	0.424	0.423	0.359	0.46	0.427
18	0.525		0.41	0.463	0.485	0.467	0.437	0.342	0.458	0.454
19	0.65		0.276	0.268	0.429	0.242	0.308	0.257	0.325	0.258
20	0.4		0.193	0.166	0.284	0.161	0.195	0.186	0.217	0.165
21	0.625		0.433	0.41	0.513	0.475	0.471	0.48	0.484	0.399
22	0.35		0.203	0.231	0.304	0.233	0.235	0.203	0.237	0.209
23	0.55		0.263	0.24	0.363	0.273	0.27	0.32	0.291	0.217
24	0.325		0.389	0.364	0.49	0.331	0.376	0.324	0.419	0.369
25	0.225		0.403	0.394	0.412	0.351	0.387	0.352	0.42	0.371
26	0.25		0.17	0.14	0.253	0.149	0.179	0.159	0.19	0.156
27	0.375		0.156	0.154	0.234	0.168	0.179	0.149	0.196	0.162
28	0.4		0.387	0.425	0.466	0.398	0.404	0.328	0.412	0.405
Correlation			0.365	0.376	0.511	0.454	0.463	0.493	0.434	0.349

K: The differences between similarity calculations in Simmetrics and SecondString

id	*Compared strings*	*Difference between Jaro metric*	*Difference between Jaro Winkler metric*	*Difference between Jaccard metric*	*Difference between Monge Elkan metric*
1.	'aqyiqs' and 'nanbwcr'	3.78E-9	3.78E-9	0	0
2.	'aqyiqs' and 'jndpeowa'	0	0	0	0
3.	'aqyiqs' and 'lylnxehgy'	1.43E-8	1.43E-8	0	0
4.	'aqyiqs' and 'rhcqazeqpz'	0	0	0	0
5.	'aqyiqs' and 'mjalxxhdvjc'	7.53E-9	7.53E-9	0	0
6.	'nanbwcr' and 'aqyiqs'	3.78E-9	3.78E-9	0	0
7.	'nanbwcr' and 'jndpeowa'	1.42E-8	1.42E-8	0	0
8.	'nanbwcr' and 'lylnxehgy'	1.5E-8	1.5E-8	0	0
9.	'nanbwcr' and 'rhcqazeqpz'	1.02E-8	1.02E-8	0	0
10.	'nanbwcr' and 'mjalxxhdvjc'	2.18E-8	2.18E-8	0	0
11.	'jndpeowa' and 'aqyiqs'	0	0	0	0
12.	'jndpeowa' and 'nanbwcr'	1.42E-8	1.42E-8	0	0
13.	'jndpeowa' and 'lylnxehgy'	5.52E-9	5.52E-9	0	0
14.	'jndpeowa' and 'rhcqazeqpz'	0	0	0	0
15.	'jndpeowa' and 'mjalxxhdvjc'	2.17E-8	2.17E-8	0	0
16.	'lylnxehgy' and 'aqyiqs'	1.43E-8	1.43E-8	0	0
17.	'lylnxehgy' and 'nanbwcr'	1.5E-8	1.5E-8	0	0
18.	'lylnxehgy' and 'jndpeowa'	5.52E-9	5.52E-9	0	0
19.	'lylnxehgy' and 'rhcqazeqpz'	0	0	0	0
20.	'lylnxehgy' and 'mjalxxhdvjc'	6.02E-9	6.02E-9	0	0
21.	'rhcqazeqpz' and 'aqyiqs'	0	0	0	0
22.	'rhcqazeqpz' and 'nanbwcr'	1.02E-8	1.02E-8	0	0
23.	'rhcqazeqpz' and 'jndpeowa'	0	0	0	0
24.	'rhcqazeqpz' and 'lylnxehgy'	0	0	0	0
25.	'rhcqazeqpz' and 'mjalxxhdvjc'	1.21E-8	1.21E-8	0	0
26.	'mjalxxhdvjc' and 'aqyiqs'	7.53E-9	7.53E-9	0	0
27.	'mjalxxhdvjc' and 'nanbwcr'	2.18E-8	2.18E-8	0	0
28.	'mjalxxhdvjc' and 'jndpeowa'	2.17E-8	2.17E-8	0	0
29.	'mjalxxhdvjc' and 'lylnxehgy'	6.02E-9	6.02E-9	0	0
30.	'mjalxxhdvjc' and 'rhcqazeqpz'	1.21E-8	1.21E-8	0	0

id	*Compared strings*	*Difference between Jaro metric*	*Difference between Jaro Winkler metric*	*Difference between Jaccard metric*	*Difference between Monge Elkan metric*
1.	'impending roll wish' and 'jolly emphasis fed'	0	0	0	0
2.	'impending roll wish' and 'guaranteed locking bottom'	3.62E-8	3.62E-8	0	0
3.	'impending roll wish' and 'invited infrastructure replacing'	5.56E-9	7.99E-9	0	0
4.	'impending roll wish' and 'late hand parsed'	2.09E-8	2.09E-8	0	0
5.	'impending roll wish' and 'cynical advert breed'	3.03E-9	3.03E-9	0	0
6.	'jolly emphasis fed' and 'impending roll wish'	0	0	0	0
7.	'jolly emphasis fed' and 'guaranteed locking bottom'	1.28E-9	1.28E-9	0	0
8.	'jolly emphasis fed' and 'invited infrastructure replacing'	8.83E-9	8.83E-9	0	0
9.	'jolly emphasis fed' and 'late hand parsed'	4.86E-9	4.86E-9	0	0
10.	'jolly emphasis fed' and 'cynical advert breed'	4.86E-9	4.86E-9	0	0
11.	'guaranteed locking bottom' and 'impending roll wish'	3.62E-8	3.62E-8	0	0
12.	'guaranteed locking bottom' and 'jolly emphasis fed'	1.28E-9	1.28E-9	0	0
13.	'guaranteed locking bottom' and 'invited infrastructure replacing'	0	0	0	0
14.	'guaranteed locking bottom' and 'late hand parsed'	8.48E-9	8.48E-9	0	0
15.	'guaranteed locking bottom' and 'cynical advert breed'	2.5E-8	2.5E-8	0	0
16.	'invited infrastructure replacing' and 'impending roll wish'	5.56E-9	7.99E-9	0	0
17.	'invited infrastructure replacing' and 'jolly emphasis fed'	8.83E-9	8.83E-9	0	0
18.	'invited infrastructure replacing' and 'guaranteed locking bottom'	0	0	0	0

19.	‘invited infrastructure replacing’ and ‘late hand parsed’	4.64E-8	4.64E-8	0	0
20.	‘invited infrastructure replacing’ and ‘cynical advert breed’	3.94E-8	3.94E-8	0	0
21.	‘late hand parsed’ and ‘impending roll wish’	2.09E-8	2.09E-8	0	0
22.	‘late hand parsed’ and ‘jolly emphasis fed’	4.86E-9	4.86E-9	0	0
23.	‘late hand parsed’ and ‘guaranteed locking bottom’	8.48E-9	8.48E-9	0	0
24.	‘late hand parsed’ and ‘invited infrastructure replacing’	4.64E-8	4.64E-8	0	0
25.	‘late hand parsed’ and ‘cynical advert breed’	2.85E-8	2.85E-8	0	0
26.	‘cynical advert breed’ and ‘impending roll wish’	3.03E-9	3.03E-9	0	0
27.	‘cynical advert breed’ and ‘jolly emphasis fed’	4.86E-9	4.86E-9	0	0
28.	‘cynical advert breed’ and ‘guaranteed locking bottom’	2.5E-8	2.5E-8	0	0
29.	‘cynical advert breed’ and ‘invited infrastructure replacing’	3.94E-8	3.94E-8	0	0
30.	‘cynical advert breed’ and ‘late hand parsed’	2.85E-8	2.85E-8	0	0

id	*Compared strings*	*Difference between Jaro metric*	*Difference between Jaro Winkler metric*	*Difference between Jaccard metric*	*Difference between Monge Elkan metric*
1.	'Beneath a toe bobs a minute circuitry.' and 'The welcome salary skips across the engineer.'	9.57E-3	9.57E-3	0	0
2.	'Beneath a toe bobs a minute circuitry.' and 'A potential drip reasons.'	0	0	1.11E-1	2.04E-1
3.	'Beneath a toe bobs a minute circuitry.' and 'The geared throughput invokes the nuisance underneath its arranged rocket.'	2.38E-2	2.38E-2	0	0
4.	'Beneath a toe bobs a minute circuitry.' and 'A graduate tax farms underneath her friend.'	0	0	8.33E-2	1.54E-1
5.	'Beneath a toe bobs a minute circuitry.' and 'A new bomb constrains the tree past the national dish.'	1.96E-2	1.96E-2	7.14E-2	1.36E-1
6.	'The welcome salary skips across the engineer.' and 'Beneath a toe bobs a minute circuitry.'	9.57E-3	9.57E-3	0	0
7.	'The welcome salary skips across the engineer.' and 'A potential drip reasons.'	5.19E-1	5.19E-1	0	0
8.	'The welcome salary skips across the engineer.' and 'The geared throughput invokes the nuisance underneath its arranged rocket.'	9.29E-9	6.34E-9	6.19E-2	1.03E-1
9.	'The welcome salary skips across the engineer.' and 'A graduate tax farms underneath her friend.'	6.38E-3	6.38E-3	0	0
10.	'The welcome salary skips across the engineer.' and 'A new bomb constrains the tree past the national dish.'	2.7E-2	2.7E-2	4.76E-3	1.01E-2
11.	'A potential drip reasons.' and 'Beneath a toe bobs a minute circuitry.'	0	0	1.11E-1	2.04E-1
12.	'A potential drip reasons.' and 'The welcome salary skips across the engineer.'	5.19E-1	5.19E-1	0	0
13.	'A potential drip reasons.' and 'The geared throughput invokes the nuisance underneath its arranged rocket.'	4.8E-1	4.8E-1	0	0

14.	‘A potential drip reasons.’ and ‘A graduate tax farms underneath her friend.’	2.8E-8	3.43E-8	1.49E-9	1.18E-8
15.	‘A potential drip reasons.’ and ‘A new bomb constrains the tree past the national dish.’	0	2.98E-9	2.48E-9	4.97E-9
16.	‘The geared throughput invokes the nuisance underneath its arranged rocket.’ and ‘Beneath a toe bobs a minute circuitry.’	2.38E-2	2.38E-2	0	0
17.	‘The geared throughput invokes the nuisance underneath its arranged rocket.’ and ‘The welcome salary skips across the engineer.’	9.29E-9	6.34E-9	6.19E-2	1.03E-1
18.	‘The geared throughput invokes the nuisance underneath its arranged rocket.’ and ‘A potential drip reasons.’	4.8E-1	4.8E-1	0	0
19.	‘The geared throughput invokes the nuisance underneath its arranged rocket.’ and ‘A graduate tax farms underneath her friend.’	0	0	4.17E-3	6.47E-3
20.	‘The geared throughput invokes the nuisance underneath its arranged rocket.’ and ‘A new bomb constrains the tree past the national dish.’	2.03E-2	2.03E-2	3.27E-3	5.7E-3
21.	‘A graduate tax farms underneath her friend.’ and ‘Beneath a toe bobs a minute circuitry.’	0	0	8.33E-2	1.54E-1
22.	‘A graduate tax farms underneath her friend.’ and ‘The welcome salary skips across the engineer.’	6.38E-3	6.38E-3	0	0
23.	‘A graduate tax farms underneath her friend.’ and ‘A potential drip reasons.’	2.8E-8	3.43E-8	1.49E-9	1.18E-8
24.	‘A graduate tax farms underneath her friend.’ and ‘The geared throughput invokes the nuisance underneath its arranged rocket.’	0	0	4.17E-3	6.47E-3
25.	‘A graduate tax farms underneath her friend.’ and ‘A new bomb constrains the tree past the national dish.’	1.27E-8	2.21E-8	3.48E-9	2.07E-9
26.	‘A new bomb constrains the tree past the national dish.’ and ‘Beneath a toe bobs a minute circuitry.’	1.96E-2	1.96E-2	7.14E-2	1.36E-1
27.	‘A new bomb constrains the tree past the national dish.’ and ‘The welcome salary skips across the engineer.’	2.7E-2	2.7E-2	4.76E-3	1.01E-2
28.	‘A new bomb constrains the tree past	0	2.98E-9	2.48E-9	4.97E-9

	the national dish.' and 'A potential drip reasons.'				
29.	'A new bomb constrains the tree past the national dish.' and 'The geared throughput invokes the nuisance underneath its arranged rocket.'	2.03E-2	2.03E-2	3.27E-3	5.7E-3
30.	'A new bomb constrains the tree past the national dish.' and 'A graduate tax farms underneath her friend.'	1.27E-8	2.21E-8	3.48E-9	2.07E-9

L: The Architecture of Vesuvius