



Mälardalen University
School of Innovation Design and Engineering
Västerås, Sweden

Bachelor Thesis

INTELLIGENT MATCHING FOR CLINICAL DECISION SUPPORT SYSTEM FOR CEREBRAL PALSY USING DOMAIN KNOWLEDGE

Filiph Eriksson-Falk
ffk13001@student.mdh.se
Fredrik Frenning
ffg12002@student.mdh.se

Examiner: Ning Xiong
Mälardalen University, Västerås, Sweden

Supervisor: Peter Funk
Mälardalen University, Västerås, Sweden

June 8, 2017

Abstract

Relevant information at the right time can be critically important for clinicians when treating patients with cerebral palsy (CP). Gathering this information could be done through the usage of a clinical decision support system with a matching algorithm that finds relevant patients. The relevancy of this information for clinicians is determined by the relevancy of the matched patients. The aim of this thesis was therefore to investigate how an algorithm that matches similar patients with CP could be improved in terms of relevancy. The goal was also to explore the possibilities of domain knowledge and temporal aspects and how they could be combined and utilized in order to improve the matching algorithm. In this bachelor's thesis, we have conducted a literature study about the domain and a domain knowledge survey. The domain knowledge survey included gathering domain knowledge through contact with an expert in the area of CP. We also implemented an algorithm using intelligent similarity measurements based on validation from experts that could accurately match similar patients according to the domain knowledge gathered. The resulting algorithm is presented through a prototype of a CDSS, which allows clinicians to select and match patients through a GUI, and including features such as adjusting weight values for different attributes. The algorithm uses patient data retrieved from the CPUP database, which is specific to patients with CP, to match with. From the CPUP database many temporal aspects could be concluded to be relevant for similarity assessment. Due to the limited scope of the thesis however, only the most important aspect was utilized. By treating this aspect as an attribute like the other domain knowledge based attributes, but with respect to other variables that affected it, a combination of temporal aspects and domain knowledge was done when identifying similar patients with CP. Using the prototype of the CDSS with the implemented algorithm could help clinicians make better informed decisions, and this leads to improved health care for children and patients with CP, which is why this thesis was important.

Contents

1	Introduction	4
1.1	Motivation	4
1.2	Limiatations	4
1.3	Problem formulation	5
1.4	Method	5
1.4.1	AI-technique	5
1.5	Ethical and Societal Considerations	6
2	Background	7
2.1	Case-Based Reasoning	7
2.2	Clinical Decision Support System	8
2.3	Clinical Decision Support System and Case-Based Reasoning	8
2.4	Cerebral Palsy	9
2.4.1	Gross Motor Function Classification System	9
2.4.2	Age relevancy in CP	10
2.4.3	Temporal Aspects in CP	11
2.5	CPUP	11
2.6	CPUP-registry	11
3	Related Work	12
3.1	Gathering domain knowledge	12
3.2	CDSS	12
3.3	CPUP	12
3.4	Watson Analytics	13
4	Gathering Domain Knowledge	13
4.1	Dorsiflexion and plantar flexion	14
4.2	Standing aid	15
4.3	GMFCS	15
4.4	Age Similarity	15
4.4.1	Tolerance	16
4.4.2	Dynamic Tolerance	16
4.5	Temporal Aspects	16
5	Implementation	17
5.1	Setting up the database	17
5.2	Retrieval of similar cases	17
5.2.1	Dorsiflexion similarity	18
5.2.2	Plantar flexion similarity	19
5.2.3	Standing aid similarity	20
5.2.4	Age similarity	20
5.2.5	Surgery similarity	21
5.3	Graphical User Interface	21
6	System Description	24
6.1	Dorsiflexion	26
6.2	Plantar flexion	27
6.3	Standing aid	27
6.4	Age similarity	27
6.5	Surgery similarity	30
6.6	Patient similarity	31

7	Discussion	33
7.1	Significance	33
7.2	Previous work	33
7.3	Goals	33
7.3.1	Addressing temporal aspects	34
7.3.2	Combining temporal aspects and domain knowledge	34
7.3.3	Improvement of identification using domain knowledge	34
7.4	Justification	35
7.5	Limitations	35
7.6	The bigger picture	36
8	Conclusions	36
9	Future work	37
	References	39

1 Introduction

Relevant information in the right time can be critically important for clinicians when treating patients. An area where this may apply are children with cerebral palsy (CP). In many cases it may be difficult to find relevant information about these patients, and it may be a process that takes more time than desired. This affects the decision-making for clinicians and makes it difficult for them to choose the right treatment. However, a clinician who gets a better decision basis can make a more informed decision.

Today, data is used in many different ways to improve diagnostics and treatment, but it is not used on an individual level in the daily work of clinicians. It is too resourcefully demanding and they would rather generalize instead. In context of children with CP however, it is important that the treatment is carefully adapted for each individual as it may have a major impact on the child's quality of life.

CPUP is a growing organization that manages registries of patients with CP in several countries. The registries could be very useful for developing a system that makes it less resourcefully demanding to adjust treatment for individuals with CP. This is, among other things, why the potential of a clinical decision support system (CDSS), for children with cerebral palsy was investigated by Anna Enbom [1]. In her work, a prototype of a CDSS was implemented with a relatively simple CBR (Case-based reasoning) search algorithm that identifies similar patients. However, the scope and breadth of her thesis did not give the opportunity to develop this part more in depth.

A potential development of this system is the introduction of detailed domain knowledge from experts in the field. Domain knowledge, combined with temporal aspects of CP, can be implemented into the CBR search algorithm, which may result in matched patients that better represent the opinions of clinicians. This would increase the potential of a CDSS system for clinicians who treat children with CP and act as a reliable and effective support in their decision-making.

Clinicians and stakeholders in the medical field of CP could potentially have use for a further development and refinement of the work made in this project. But the research that was conducted in this thesis could also prove to be useful for anyone who sees the benefits of domain knowledge. Software developers working with CDSSs could potentially develop this project further or see this work as related to theirs. By seeing the benefits of CBR one could adapt this work to their area or use domain knowledge in conjunction with other AI-techniques.

1.1 Motivation

Improving the health care for children in particular is something the medical field should prioritize. Children are the future and it is important to state that children diagnosed with CP are of equal value as anyone else. By helping clinicians make more informative decisions we hope to increase the function of children with CP and thus improve their quality of life.

Domain knowledge is not only interesting from a healthcare perspective, but it could also improve search algorithms in several types of areas, such as search engines online [2]. This is why domain knowledge is a relevant topic and why the connection between domain knowledge and search algorithms is worth investigating more.

In addition to domain knowledge, temporal aspects is also an interesting area to explore in this system. By investigating this area and combining it with domain knowledge, the system would be able to bring new aspects in to matches and analyzes. This is something that could be useful for clinicians, since it would mean that the system could allow them to analyze similarities between patients in new ways.

1.2 Limitations

This thesis focuses on CDSSs, CP, CPUP and CBR, which are four large subjects with a lot of research potential. But since this thesis work were under strict time limitations, the scope of the project had to be narrowed down. To clarify what the focus is of this thesis work, we are not going to create a complete CDSS using CBR. The focus is on the similarity measurement between patients, which is done through developing an algorithm that represents only a part of the CBR-process.

Because of the narrowed scope of the project, the complete CPUP-registry will not be investigated in this work. The research will be restricted to the most important attributes of the CPUP-registry of when dealing with similarity measurements between patients with CP.

1.3 Problem formulation

The aim of this thesis is to explore the possibilities of domain knowledge and temporal aspects in similarity assessments between patients with CP. The purpose is also to investigate how, through the use of domain knowledge and temporal aspects, matching of similar patients with CP can be done through CPUP data in a similar way that experts would have done.

The research questions that will be answered in this thesis are the following:

1. How can identification of similar patients with CP be improved using domain knowledge?
2. How are temporal aspects addressed when matching patients in the CPUP registry?
3. How can you combine temporal aspects and domain knowledge when identifying similar relevant patients with CP?

1.4 Method

The work consists of a theoretical part and a practical part. The theoretical part includes a domain knowledge survey and a literature study. The literature study was conducted through searches in established databases such as IEEE Xplore, Springer, PubMed and ACM. The searches aimed to find information about CP, CPUP, CDSS, temporal aspects and domain knowledge. The domain knowledge survey consisted of an interview with an expert in the field of CP and CPUP, which was done to gather relevant information about domain knowledge and temporal aspects of CP. In addition to the interview, we took part of the domain study performed by Enbom [1] to get started with the work quickly and avoid redundant work. Her domain study describes a bit about how to interpret the CPUP registry (described in section 3.5 in her thesis). This made the interpretation of the database and its large amount of attributes less resource-intensive to process in our work. The theoretical part also included the verification of significance each attribute were to have in the algorithm with experts.

The practical part of the work involves the implementation of an algorithm that intelligently matches patients with CP using domain knowledge and temporal aspects. The implemented algorithm is in the form of CBR and uses k Nearest Neighbour (k-NN). The reasons why CBR was the algorithm of choice for this work is explained in section 1.4.1. Domain knowledge, together with temporal aspects, are used in the system primarily by influencing the algorithm's feature selection, weights and similarity calculations. In conjunction with the implementation, we gathered data from the CPUP registry that were used to test real data.

Since it was ultimately intended that the algorithm was to be used in a CDSS, we reused and adjusted Enbom's [1] prototype of one since it seemed appropriate. Through the GUI used in her prototype, which is in the form of a computer program developed in Java, it was easier to see the resulting matched patients' information, hence more easily being able to determine the relevance of the matched patients.

1.4.1 AI-technique

Based on the requirements set to solve a problem, established AI techniques often differ from each other to solve different problems and achieve different goals. In order to choose a technique that was suitable for solving the problem in this thesis, the existing complications had to first be understood. The most important criteria that the AI techniques had to meet in this case was how well they could operate with the CPUP registry. It was also important to investigate the strengths, limitations, and demands that the techniques may have had. It was critical to state that the algorithm was not searching for solutions but rather relevant information. Anna Enbom has analyzed several AI techniques, and her results show that Case-Based Reasoning was clearly the most suitable approach for the purpose [1]. Some important requirements, according to Anna Enbom, which help distinguish the most suitable technique are:

- The algorithm should not be black box since clinicians probably want to validate the results before making decisions
- The availability of expert knowledge is very limited, which excludes rule-based solutions
- The algorithm should be able to handle uncertainties in the data, even when clinicians find it hard to explain how different values relate

Based on these points, the conclusions were that Artificial Neural Network (ANN) was not appropriate because of it being a black box implementation. Genetic Algorithm (GA) manages feature selection well, and since we aimed to obtain more detailed domain knowledge and explore how it could be integrated into the system, GA would be able to accommodate our case. A critical disadvantage however, was that GA generally requires a lot of data to generate good results. From this point of view, the CPUP registry was too small to support GA. Support Vector Machine (SVM) and Decision Tree could also work, but that would first require the data to be converted, and the data is not structured in a way for this type of technique to be beneficial. The advantage of CBR was that the algorithm does not require large amounts of detailed domain knowledge and it was not a black box solution.

The way CBR works is similar to how people handle problem solving in daily life [3]. When a clinical decision is complex, it is common to ask more experienced colleagues for advice [4]. Since CBR resembles a reasoning process that people are used to and find intuitive, it was more likely that clinicians would accept it [5]. Another advantage of CBR is that it can present many options for the same problem. Problem situations arising when choosing the right treatment for patients are many, since a certain treatment that works for most patients with a certain condition may not work for another patient with the same condition. A clinician could then, instead of receiving a recommendation that is the statistically best case, be presented with both the majority of cases where treatment has worked and the more unusual cases where other treatments have worked better. Based on the reasons mentioned, we therefore concluded that CBR is most suitable as AI technique in this work.

1.5 Ethical and Societal Considerations

In the general area of health care there exists information that is considered sensitive and it is important to deal with this information carefully. Key identifying attributes such as name and social security number are considered as sensitive information and these should not be published online without the consent of said person. The CPUP database contain large amounts of sensitive information about patients, however, the parts of the CPUP database we accessed first had this information removed, and by signing a non-disclosure agreement we guaranteed that no sensitive information would intentionally be included in this report.

The information gathered from the interview would not be connected to the person itself unless the person agreed to, and the person would be, if requested, guaranteed to be anonymous in the report by writing an agreement. Any sensitive information collected through the interview would be excluded from report since it was not the goal of the interview. The algorithm and CDSS would use data from CPUP database but this data would also be stripped of sensitive information.

Another important topic to address is how philosophically different the view of how perfect an initial software product is expected to be by physicians and software developers. Branningan and Dayhoff [6, p. 13-14] explain the philosophical differences between software developers and physicians accordingly: "Physicians expect perfection from themselves and those around them. Physicians undergo rigorous training, have to pass multiple licensing examinations, and are held in high esteem by society for their knowledge and skills. In contrast, software developers often assume that initial products will be "buggy" and that eventually most errors will be fixed, often as a result of user feedback and error reports." This meant that there could have been different expectations, by physicians and us, of how perfect the prototype and the algorithm in this thesis was to be. Since a CDSS in some ways functions like a clinical consultant, these differing expectations could prove to be problematic, especially if the reasoning mechanism of the CDSS is not transparent to the user. Writing an extensive user manual was out of the scope for this project, but the information written in this report should, however, be sufficient to understand the basic limitations and strengths of the developed algorithm.

2 Background

In order to develop a new algorithm that combines several different concepts, thorough knowledge about these are needed. The algorithm is being developed with the aim of ultimately improving the quality of life for people with CP, by supporting clinicians with their decision making through a CDSS. Therefore, knowledge about these systems and disabilities is also required. Section 2.1 describes CBR and k-Nearest Neighbour (k-NN). How a CDSS works in healthcare is described in section 2.2. The integration of CBR in a CDSS is described in 2.3. CP is described in Section 2.4, along with GMFCS, age factor and temporal aspects. Knowledge about CPUP and its registry is necessary to understand how data is to be applied. CPUP is therefore described in Section 2.5 and its database in Section 2.6.

2.1 Case-Based Reasoning

Case-based reasoning (CBR) is a way to solve problems by reusing information and knowledge from previous experiences [7]. A case in CBR is a previous experience of a problem situation. For example, a case in a CDSS would be: "a patient record structured by symptoms, diagnosis, treatment and results" [5]. A "case base" is a collection of cases. The purpose of the case base is to be used to solve future problem situations.

In order to define a case according to the most basic of forms, one only needs a problem and a solution [3, p. 20]. In more detailed forms, a case may also contain other information, such as useful reasonings, how often the solution has been used and how successful it has been. For CBR to be operational it requires problems and solutions, but usually no information about how the problem was resolved is required.

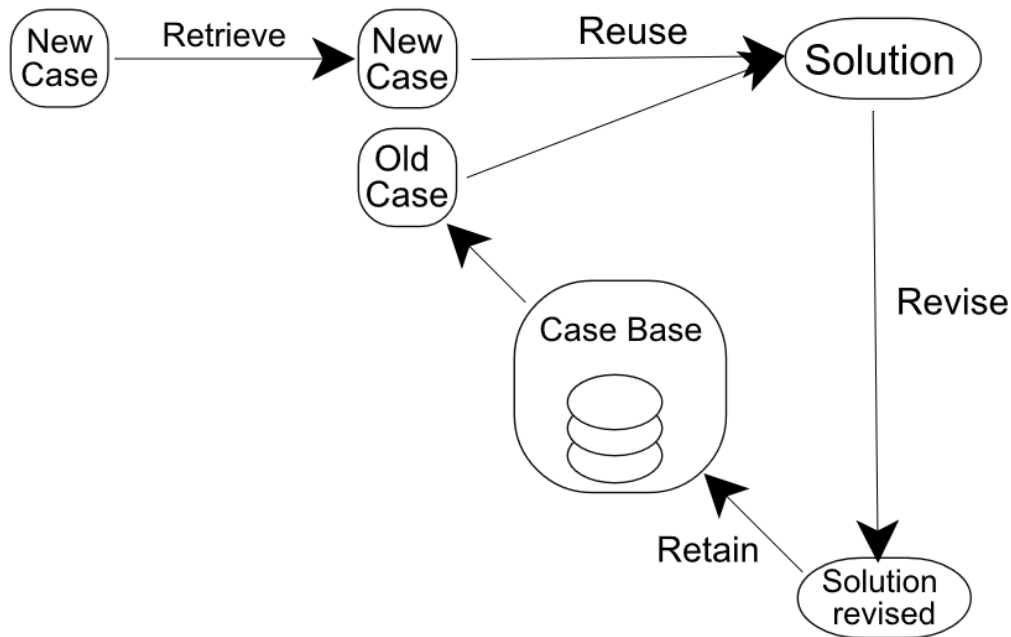


Figure 1: The four phases of CBR.

CBR processes are usually divided into four phases, as explained by Richter et al. [3]. An illustration of these phases can be seen in figure 1. The first is the retrieve phase, in which one or more previous cases are retrieved similar to the current problem. Two cases are considered similar if they have the same solution with exception for minor differences [3, p. 114, 201]. This means that the problems do not have to be exactly the same, as long as the solutions are similar.

The second phase is the reuse phase where cases retrieved during the retrieve phase are reused to propose a solution to the current problem. While the retrieve phase usually focuses on the problem part of the case, the reuse phase focuses on the solution [3, p. 30]. To reuse a case, the

easiest way is to take the case as it is, but usually some modification is necessary in order for it to be properly adapted to the current situation. The modification takes place either manually or automatically, and rule-based methods are usually preferred when entering the information used to modify the case. It is also possible in the reuse phase to reuse the strategy used to develop the solution [3, p. 190].

The third phase is the revision phase (revise), and in this phase the proposed solution is evaluated. Should any errors be found, they will be repaired as much as possible.

The last phase is the retain phase. This is where cases, which can be considered useful in the future, are stored in the case base [3, p. 223]. What makes CBR special is its way of handling specific data from previous problems. Other AI techniques usually use more general knowledge from the problem area and the relationships between different problem descriptions and conclusions. Mistakes as well as successful cases can be retained because it is possible to learn from mistakes too. This makes CBR an incremental technique where each new case is stored and helps expand the technique.

CBR requires a large enough case base. But even with an extensive case base, there may be a shortage of cases that match patients with unusual diseases. On the other hand, if there is a well-defined domain knowledge, artificial cases can be created [8]. Although it may be necessary to have an arbitrarily large amount of cases, it comes with the need for more storage and the search time increases.

Feature selection and weighting mechanisms are two key elements that CBR uses to match similar cases. These factors often integrate with domain knowledge to best adapt the matching in the retain phase to the specific area in question [9].

When similar cases have been found, CBR is combined with search algorithms like k-NN for selection. K-NN is a machine learning technique and is used in two cases: classification and regression [10]. In both cases, k-NN takes k points of the nearest objects from the training examples as input. In classification, k-NN appoints a class to an object and the class is generated through a voting system. It is k of the nearest neighbours who vote and through majority the object's class is decided. The class is the same as the class of the nearest neighbours.

2.2 Clinical Decision Support System

With the increasing flow of medical knowledge and ever since the transition to electronic medical records, clinical decision support systems (CDSSs) have been established as necessary tools for clinicians and stakeholders in the medical profession [11]. The purpose of a CDSS is to improve patient care by providing relevant information for clinicians. The process that provides this information involves matching of the patient characteristics to a knowledge base [12]. The patient characteristics can be entered into the computer system through several ways. It can, for example, be retrieved by querying the patient medical records; or be manually entered by practitioners, health care staff, or by the patients themselves. Software algorithms then generate patient-specific recommendations, which are delivered to the clinician through, for example, the medical record, printouts placed in a patients paper chart, or by pager. An example of a CDSS architecture that is similar to this structure was presented by Robbins et al. [11], which is illustrated in figure 2.

The authorized knowledge base is updated continuously by, among other things, the scientific community through journal and conference publications. In turn, the patient model and treatment library is updated by the knowledge base after the information has been validated by clinicians. The patient model determines what information is stored about a patient, which can be interpreted as its characteristics. The treatment library presents proper treatment after diagnosis. Intelligent agents helps with handling of data and matching, which can be interpreted as the part where patient-specific recommendations are generated by software algorithms. These four parts are illustrated in figure 2.

2.3 Clinical Decision Support System and Case-Based Reasoning

Normally, a clinician will receive several suggestions from the CDSS to look through, from which the clinician can extract useful information and discard non-useful information [6, p. 4-5]. There is no restricted way of generating these suggestions, but one way is to use software algorithms that

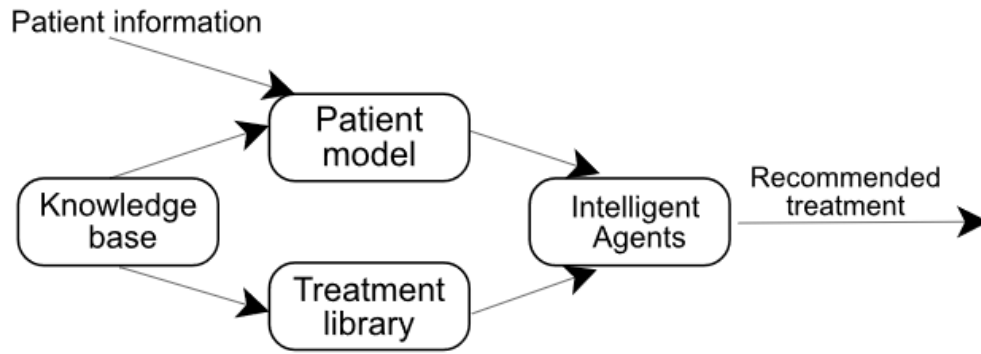


Figure 2: A CDSS architecture presented by Robbins et al. [11].

calculate the relevant suggestions according to specified rules. This is called a rule-based system, where each rule could be, for example: "IF a patient has a hard time standing AND IF the patient uses a wheelchair THEN an orthosis treatment is recommended" [6, p. 5]. The knowledge base would in this case consist of these rules. A system to introduce these rules into the knowledge base by clinicians themselves was developed by [13].

A CBR algorithm can be integrated into a CDSS naturally as it includes the process of storing cases of patient records, matching of similar cases, and presenting the similar cases as suggestions. The algorithm does not have to use a rule-based system for finding similar cases however, as it is more common to base the similar found cases on previous cases in CBR. Using this method, it is more convenient and realistic in the domain of CP, since it is not easy to state which treatment a patient with CP needs based on rules.

2.4 Cerebral Palsy

CP is a group of permanent movement disorders which a child may be affected by through brain injury somewhere between pregnancy to the child's second birthday. There are about two children in 1000 who are afflicted by CP in Sweden and the disabilities from CP varies in strength [14]. By affecting muscles in different ways, one patient may experience major mobility impairments from the disorder while another may only experience minor impairments. With proper treatment, support and training one can reduce the problems related to CP and as an adult one can live close to a normal life including education and work. But to make a decision about what treatment should be given to a specific patient, a clinician base it on several crucial factors. Three of those are GMFCS-level, age and temporal aspects. To give a bit more insight about why these are important, we have described them in more detail in the following sections. Section 2.4.1 describes GMFCS, section 2.4.2 describes age and section 2.4.3 describes temporal aspects of CP.

2.4.1 Gross Motor Function Classification System

GMFCS was introduced 1997 by Palisano et al. [15] to fulfill the need of a standardized classification system of gross motor function for patients with CP. Prior to GMFCS the research regarding CP had identified the connection between motor outcome and severity of the disorder, however, there was no standardized classification of motor disability system that was recognized or used in clinical practice and research. One of the reasons and difficulties behind establishing this sort of standardized system was that experts weren't unanimous in the definition and distinction of gross motor function, but many still believed a standardized system was needed. Palisano et al. [15] presented the drafts of the GMFCS for groups of experts in order to validate the content and it was done through nominal group process and Delphi survey consensus methods. During the Delphi survey, twenty-one experts (developmental pediatricians, physical therapists and occupational therapists) was gathered to review and validate the content of the GMFCS levels, and reaching consensus in the matter was hard. The experts had different views on the distinction of what each level represented regarding motor functionality. The five levels can in a basic form be listed as:

- **GMFCS Level I:** Walks without limitations
- **GMFCS Level II:** Walks with limitations
- **GMFCS Level III:** Walks using hand held mobility aid
- **GMFCS level IV:** Self-mobility is limited; may use powered mobility
- **GMFCS Level V:** Transported by manual wheelchair

After the survey, the interrater reliability [15] was reviewed for the classification system with the kappa (k) coefficient used to correct chance agreement. The review pointed out that younger patients had lower k value meaning the experts were less similar in their definition of motor function in those cases. The interrater reliability analyzes were performed separately for each age group and in addition to the general level of chance-corrected agreement, specific (K) values focused on help determining the five levels of classification was derived for the system. The conditional probability was then calculated to evaluate category associations and this was done through two cases where one randomly selected therapist choose a particular level of function, and then a second therapist choose all five levels of function.

When it comes to categorization in GMFCS, patients are divided in different age bands and one age band is defined between ages 6-12. Patients in the 6-12 age band are called children while in the age band between 12-18 patients are called youth and the factor that separates age bands are the content of each level the GMFCS contains. This means a child with GMFCS level III may not have the same symptoms as a youth with the same GMFCS level. The differences, however, are minor and in most cases they don't matter. To separate a patient between the levels of an age category, certain distinctions have been specified. The distinctions between, for example, levels III and IV for children and youth are [15] "Differences in sitting ability and mobility exists, even allowing for extensive use of assistive technology. children in level III sit independently, have independent floor mobility, and walk with assistive mobility devices. Children in level IV function in siting (usually supported) but independent mobility is very limited. Children in level IV are more likely to be transported or use powered mobility."

In order to further validate the content of GMFCS, Palisano et al. [16] performed a study eleven years after the introduction of GMFCS. Once again using consensus methods, a group of 18 experts was gathered to draft the GMFCS levels for the age band of 12-18 and evaluate the results once more.

The work made by Palisano et al. [15] lead to reliable results and GMFCS was established as a standard for classification of motor function for patients with CP. Using consensus methods shows that no result was forcefully applied and clinicians can use GMFCS as a trustworthy tool.

2.4.2 Age relevancy in CP

Examining patients with CP leads to information that can be rather specific to a patient or a symptom. Some patients have hip disorders and others have knee disorders, what they all have in common is a level of GMFCS and age. The age factor is very relevant for clinicians when examining a patient from several perspectives. One perspective is how old the patient is and what sort of risks or common symptoms that relate to that age. Another perspective is the development of a symptom over time and this perspective was adopted by Hägglund et al. [17] in their examination of the development of spasticity (a common subtype of CP) for children with CP. "The children with spastic unilateral CP showed an increasing muscle tone up to 4 years of age, followed by a decreasing tone up to 11 years of age. After 11 years of age an increasing muscle tone was registered when children operated with TAL were included. When those operated with TAL were excluded, no statistically significant change in muscle tone was seen after 7 years of age" (TAL stands for Tendo Achilles Lengthening which also is a subtype of CP.) This shows that symptoms of CP varies over time and depending on the age of the child, certain symptoms are more likely to appear.

2.4.3 Temporal Aspects in CP

Temporal aspects can be many within CP because patients with disabilities differ a lot, both in symptoms and treatments. If a patient performs surgery at the age of 12 and another patient at the age of 18, these could be considered as two temporary events. What makes them interesting is what the outcome the surgeries is and if there are similar effects such as a 30% improvement five years after surgery. PESMiner [18] is a tool designed to find patterns in large interval-based amounts of temporal data. According to the authors [18], there are three major challenges dealt with in this type of analysis. The first is to discover unknown and unexpected patterns and thus extract new knowledge from them. The second challenge is to respond to the need for storage and computing capacity large amounts of data requires. The third revolves around the complexity of large-scale data sets, which means that patterns can't be completely predetermined. Therefore, researchers and experts are required to test results during the process and to enter newly discovered knowledge into the algorithm for the next iteration.

2.5 CPUP

CPUP is a follow-up program for children with CP and has been a national quality register in Sweden since 2005. It was founded to prevent complications such as hip fluxes and contractures by observing children in a structured manner. The CPUP follows the results of continuous treatment efforts from infant age up to adulthood. The purpose of this is to have the opportunity to provide the right treatment at the earliest stage to prevent hip fluctuation and difficult contractures. With proper treatment on time, a patient's motoric function and quality of life can be improved. The CPUP system also aims at increasing knowledge about CP and improving collaboration between different occupational categories around people with CP. It has been proven that through the CPUP, it is possible to prevent hip fluctuation and reduce the number of children who develop difficult contracture and skolios [19].

Initially CPUP was only aimed for children and adolescents, but since 2011 adults are also included in some parts of Sweden [20]. In 2015 there were a total of 4545 participants in the CPUP in Sweden, of which 3716 were children and 828 were adults [21]. Sweden is perhaps world leading in collecting data on children with CP, this can be concluded from the fact that only five percent of children with CP in Sweden are missing from the database. The number of children in the CPUP registry corresponds to over 95% of all children and adolescents with CP in Sweden.

The CPUP is expanding every year, and as of today there are programs throughout Sweden, Norway, Denmark, Scotland and parts of Iceland and Australia [19].

2.6 CPUP-registry

The CPUP registry contains a large amount of information about patients with CP. Data in the CPUP registry comes from X-rays and examinations made by physiotherapists, occupational therapists and neural pediatricians [22]. The most interesting part for this project is probably data from physiotherapists examinations of children.

Children with relatively good gross motor function and ability to use handheld tools are examined once a year to 6 years of age, and then every other year. Children with less gross motor function are examined twice a year to 6 years of age, and then once a year. Adults are examined either every year, every two years or every three years depending on their gross motor function. Additional examinations can be performed if a physiotherapist considers it necessary [22].

3 Related Work

There are several other researchers who has gathered domain knowledge regarding CP. However, in terms of using it for similarity measurements between patients, the research is very limited. The work closest related to ours is the research made by Anna Enbom in her master's thesis [1]. Enbom conducted a domain study including an interview with a physiotherapist about which attributes in the CPUP registry are among the more important to use when matching similar patients with CP. Based on these attributes, Enbom developed a prototype of a CDSS using a CBR algorithm that used the attributes to calculate the similarity between patients with CP. However, the algorithm that she used in her CDSS prototype was based on hard-coded values and weights for matching, which was because of limitations in the scope of the thesis. This is where our work separates from Enbom's, since our algorithm includes more generally applying domain knowledge, temporal aspects and utilize intelligent and dynamically adapting values.

Because of the limited research regarding our specific area, the following sections presents related research about some of the major parts of the work in this thesis.

3.1 Gathering domain knowledge

Research regarding key identifying characteristics of CP is extensive, the research made by Steven et al. [23] focused on the persistence of CP diagnosis over time. They examined patients with and without CP and divided patients with CP in subgroups of disabling CP and nondisabling CP. Their results show that some patients resolves their CP classification due to uncertain CP status at the first diagnosis. Their research also show that at certain ages some features of the disorder are more present than other ages, which was presented through statistical evidence like the following: "At age 6, 90% of children with persistent disabling cerebral palsy needed help eating or bathing, 79% had trouble walking stairs, and 54% were unable to walk unassisted; at age 9, 44% used a wheelchair and 18% used crutches."

3.2 CDSS

In an article about a CADSS (computer-aided decision support system) developed for usage in the field of endoscopy [24], the authors differentiate automated decision support systems from systems that focus on content-based image/video retrieval (CBIR/CBVR). The output generated from automated systems is usually processed without any intervention from a medical expert. CBIR/CBVR systems allow experts to decide for themselves what the final diagnosis should be. This is done by presenting a number of similar images or videos for the expert of the diagnosis and let them choose the best alternative. The expert is usually allowed to refine the query by interacting with the system and search for similar images. Therefore, CBIR/CBVR systems are limited in usage of real-time appliances and they are restricted to offline processing.

"An increasing number of decision support systems based on domain knowledge are adopted to diagnose medical conditions such as diabetes and heart disease." A citation from [25] in a paper about fuzzy ontology. The authors [25] propose a fuzzy expert system aimed at resolving data and uncertain knowledge problems with diabetes. They implemented a SDSA (semantic decision support agent), used for diagnosing diabetes by semantic decision making, using an experimental data set to generate fuzzy concepts and fuzzy relations. By associating fuzzy relations with a layer of fuzzy diabetes concepts using fuzzy numbers, a decision mechanism execute rule-based inference to extract the possibility of a patient being diagnosed with diabetes. The results generated from their experimental data indicate that their method can analyze data and simulate human thinking process.

3.3 CPUP

Experiencing pain is a complication common for patients with CP and with the need to investigate how present the pain can be, population-based studies was performed by [26]. Using the CPUP database, Alriksson-Schmidt et al. extracted variables such as age, GMFCS and gender to use as narrowing factors of identifying why and when pain occurs. The results of their research showed that reporting pain was more frequently done by females and that the amount of reports of pain

increased with age. Patients with GMFCS level of III was more likely to report pain than those with a GMFCS level of I. CPUP does include almost all children with CP but it doesn't explicitly provide an attribute representing reports of pain, and this was addressed as an issue by Alriksson-Schmidt [26]. Alriksson-Schmidt [26] expresses the risk of including a higher selection of patients without presence of pain in their investigation by using CPUP.

The information gathered by the CPUP surveillance program aims to improve quality of life and prevent symptoms such as severe contractions for patients with CP. However, the data stored in the CPUP database can be used for a lot more than improving the quality of life for children with CP. A study investigating survivability for children with CP, using data from CPUP, was conducted by Westbom et al. [27]. The research aimed to find cause of death for children diagnosed CP up to ages of 19 and investigate if the cause could relate to the CP disorder. Death may seem unlikely for patients diagnosed with CP, at least where the CP disorder being the sole reason, and the results from Westbom et al.'s research was that 4% of the patients included in the study died before 19 years of age. The authors also concluded that patients with GMFCS level V had 60% chance to survive 19 years or older and patients with higher function had significantly higher chance of survival. The authors also state that even though the CPUP database was used to find patients, no bias was suspected due to systematic drop-out.

3.4 Watson Analytics

A software tool that is getting more popular to use for gathering information about data is Watson Analytics [28]. It is a tool that analyzes and visualizes data to provide insight about it. It discovers information about data by guiding the user in exploration of it. The company behind the creation of this tool, IBM, have recently made an online API available to use for free, which has made it more common to think that this tool will be able to replace decision support systems altogether [29]. But even if this tool would be able to replace some decision support systems, it would not be able to replace those that are based on detailed domain knowledge. As mentioned by [30], Watson Analytics finds results based on how statistically "interesting" they are, and makes suggestions for how to best tackle a problem after that. If a patient with CP and specific symptoms needs surgery or not is a decision based on many parameters that can not be simply explained by statistics. It would need proper domain knowledge with a system that does not present to the user one clear "best" decision, but rather provide relevant information about the patient that could help clinicians make informed decisions. Watson Analytics could be useful for clinicians in other ways however. In a paper by [31], they used Watson analytics to summarize Electronic Medical Records (EMRs) of patients, with the purpose of making EMRs easier for clinicians to interpret and find relevant information in. In this work, Watson analytics was used to visualize the different parameters of EMRs, such as notes and historical values of medications and labs.

4 Gathering Domain Knowledge

To be able to gather domain knowledge about CP and CPUP, we contacted Elisabet Rodby Bousquet from "Center for Clinical Research, Region Västmanland". She agreed to an interview, to discuss and point out the crucial factors that can be used to define similarity between two patients with CP. Together with a literature study (described in section 1.4) about the domain, which results are described in section 2.4, and by reusing some of the information gathered in a domain study made by Enbom [1], we were able to prepare relevant questions for the interview. The interview with Elisabet was held in person, where we got to ask her questions regarding which attributes we could use from the CPUP registry, but also if there were other aspects that may affect similarity. We also had continuous contact with her through email after this, for the purpose of validation and answering questions that arose throughout the work.

Some of the attributes we chose to discuss with Elisabet were the ones Enbom [1] had specified as more significant in her thesis. These were supposedly related to ankle movement and were investigated for relevance in her work. However, due to the scope of her project, sufficient domain knowledge about these attributes could not be gathered by Enbom to include all of them in the matching of similar patients. The investigated attributes by Enbom were all relevant in different ways according to Elisabet, but there were also three other attributes from the CPUP registry that

she thought might be worth including. These were regarding whether a patient uses standing aid or not, how long he/she uses it per day, and how many days per week he/she uses it. It was also to be noted that every entry in the CPUP-registry represented an examination occasion of a patient, which meant that every attribute was supposed to be connected to an examination. There were however some columns that were connected to separate occasions, like surgeries for example. This meant that there were large amounts of data stored in the registry to process.

Other than the attributes from Enbom’s work, we also discussed temporal aspects of CP and how we could integrate these in the similarity measurement. This was concluded to have potential of being an interesting part of the algorithm by investigating significant treatments in the CPUP-registry. The information we gathered about temporal aspects will be explained further in section 4.5.

There are a large variety of attributes included in the CPUP-registry, but since we had to narrow the scope for this project due to the strict time limitation, we chose to focus on the attributes that were considered most important according to Elisabet. These attributes are listed in table 1, and the following sections explain why each attribute is important for similarity measurements between patients with CP. The sections will also describe how these attributes could be implemented into a matching function, as explained by Elisabet.

Attribute	Type of value
Age	Year and month
GMFCS	Level 1-5
"Uses standing aid"	Yes/no
"Uses standing aid: hours per day"	4 alternative ranges: ≤ 1 , $1 - 2$, $3 - 4$, ≥ 4
"Uses standing aid: days per week"	4 alternative ranges: 1-2, 3-4, 5-6, 7
"Ankle joint: dorsiflexion at flexed knee - right"	Degrees
"Ankle joint: dorsiflexion at flexed knee - left"	Degrees
"Ankle joint: dorsiflexion at extended knee - right"	Degrees
"Ankle joint: dorsiflexion at extended knee - left"	Degrees
"Plantar flexion - right"	Level 0-4
"Plantar flexion - left"	Level 0-4

Table 1: Important attributes according to Elisabet.

4.1 Dorsiflexion and plantar flexion

The range of motion for dorsiflexion represents the angle that an ankle joint can be bent upwards. Elisabet explained that it is measured by physiotherapists at examinations for both right and left ankle joint, and they measure the angles for when the knees are straight or bent. This meant that there were four attributes to be matched: dorsiflexion left and right; dorsiflexion with extended knee left and right. A relevant thing we noticed with these values was that they were supposed to be represented as rough estimates. According to Elisabet, if an angle in the database was registered as, for example, 4 degrees, it is most certainly a measurement error since it was then supposed to have been entered as a rough estimate of 0 degrees. This was important to note because reaching the 0 degree-mark is critically important for patients, since that would mean that they could have the foot in a perpendicular angle, which means that they can stand with the whole foot on the floor. What we could do to calculate a similarity based on this was not very clear however, but it was not to be used as a strict threshold that separates values below 0 from values above 0.

A significant difference between two values of dorsiflexion was about 20 degrees, but no details of other significant differences were concluded. Dorsiflexion was an important attribute since it could represent the outcome of treatments, which meant that the amount of weight dorsiflexion should have was quite high. It should not have more of an impact on the total similarity between patients than age and GMFCS do however, according to Elisabet.

Plantar flexion in the CPUP-registry represents the spasticity of the ankle joint, which is measured by the amount of muscle tension there is. The attribute value ranges from 0 to 4 in the

form of integers (or levels), where 0 represents no elevated muscular tension, and 4 represents that the ankle joint is complete stern and can not be bent. No suggested way of matching this attribute was concluded by Elisabet, so we made a suggestion of subtracting 25% similarity-value for each step of level difference. The suggestion was approved by her as it seemed like a wise solution.

The attributes for dorsiflexion and plantar flexion are appropriate to use for finding similar patients in the CPUP registry for several reasons. The most significant reason is that they has been entered consistently into the registry for a long time, which means that there exists a lot of data to match with. It is also appropriate to use these attributes as a measurement of how good or bad treatments worked for patients, since clinicians could easily interpret the alteration of these attributes over time.

4.2 Standing aid

Standing aid is registered into the CPUP-registry as three columns, which can be seen in table 1. When calculating the similarity of the attribute "Standing aid", which is a yes or no question, it could simply be done by strictly giving 100% similarity if both patients has the same value, according to Elisabet. For matching the attributes "Uses standing aid: hours per day" and "Uses standing aid: days per week" however, there were no clear answer of how much tolerated difference there could be for them to be matched with a certain amount of similarity-value. We made a suggestion about multiplying the two attribute values to calculate a value that represents the amount of usage of standing aid in hours per week. We thought that this would reduce the amount of similarity calculations needed since both attributes describes usage of standing aid but in different units. The suggestion was not verified by Elisabet because of lack of time for this project.

The weights of these attributes could not specified precisely, but they were not to be as heavy as age for the total similarity between patients, according to Elisabet.

4.3 GMFCS

A filtration of patients could be done based on the GMFCS value, according to Elisabet. Since each grade of GMFCS differs a lot in terms of symptoms and the treatments needed for a patient, it was critical that only patients with the same GMFCS value should be matched.

4.4 Age Similarity

Age was a factor that had great impact on the similarity between patients with CP. As explained by Elisabet, the reason for this was that the human body is developing at a fast pace as a child, which makes a child respond differently to treatments for each year that passes by. The thought process of treatment decisions made for a child of age 8 should thereby be seen as very different compared to a child of age 12 for example. How much tolerated difference in age between patients for them to be considered similar was depending on how old they were. The younger a patient was, the smaller room for difference there would be.

A treatment will most likely give a completely different result in long term for a child at a young age than for a child who is several years older. This is why age is such an important factor to consider when matching patients. But it is not easy to classify how much of a difference in age should be tolerated at a specific age, according to Elisabet. We were given a few guidelines from her where she roughly classified age groups that defined which ages could be matched together. These age groups are shown in table 2.

Age group (years)	0 - 5	6 - 7	8 - 9	10 - 12	13 - 15	16 - 18
-------------------	-------	-------	-------	---------	---------	---------

Table 2: Roughly classified age groups

To accurately classify age groups for the purpose of similarity assessment was hard since it was not something that could be done with 100% certainty, and this was because there were, to our knowledge, no proven and clear answer for it. It was concluded that what is seen as relevant to a

clinician when matching patients by their age should probably not be seen as a hard line between, for example, a 12-year old and a 11-year old. Especially if, say, the 11 year old turns 12 in the next month or so. This was made clear by Elisabet, and with this in mind, we made a suggestion to introduce the classification of ages into our system in a more intelligent manner. The parts of this suggestion is explained in sections 4.4.1 and 4.4.2. The suggestion was approved by Elisabet, as it would most certainly prove to be a more realistic approach of calculating similarity-values for the age of patients.

4.4.1 Tolerance

The first part of our suggestion was to include tolerance-values for each age group, which would represent the maximum tolerated age difference for two ages to be of 100% similarity. For example the age group of 8-9 could have a tolerance-value of 1, and the age group 10-12 could have a tolerance-value of 2. This would mean that a patient of age 9 would still be able to be matched with a patient of age 10, even though they are in two different age groups. A problem with our suggestion, however, was that a 11 year old will be able to be matched with a 9 year old, but not the other way around. This problem is explained further and solved in section 5.2.4.

Another problem was that a 9 year old would be able to be matched with a 10 year old at maximum, even though the 9 year old may have only been a week away from turning 10 years old. This is a problem since the next week, when the child would become 10 years old, it would suddenly be able to be matched with a child of age 12. This was too harsh of a threshold for it to be realistic, which leads to the next part of our suggestion that is explained in 4.4.2.

4.4.2 Dynamic Tolerance

Dynamic tolerance would mean that the tolerance-values does not strictly stick to integers for each age, but rather dynamically increase or decrease depending on how close the age was to another age group. If an age is 9 years and 11 months old, the tolerance-value should be closer to 2, which is the tolerance when reaching an age of 10. This would mean that an age close to 10 years would nearly be able to match with 100% age-similarity to an age of 12 years, since that is what an age of 10 would be able to do. With this in mind, Elisabet made new age-group classifications regarding which tolerance values that should be connected to each age-group. These can be seen in table 3 below.

Age group (years)	Tolerance (years)
0 - 4	0.5
5 - 10	1
12 - 16	1.5

Table 3: Classification of age groups by tolerance values

The problem mentioned in section 4.4.1, of an 11 year old being able to match with a 9 year old (and not vice versa), still remained though, but in this case for the ages of e.g. 5 and 4. The problem even became more complicated because of the dynamic values needing to be adjusted for both an increase and a decrease in age difference. The solution we came up with for this problem is explained in section 5.2.4.

4.5 Temporal Aspects

Temporal aspects was concluded to be mostly major events such as surgeries, according to Elisabet. Surgeries are rare and the majority of surgeries have a long lasting and often drastic effect. Due to the rarity of surgeries and the high impact that they have, Elisabet saw it appropriate to group surgeries as an own attribute for similarity matching. The way we could match this was by looking at how old two patients were when the surgeries were made and base the similarity on the difference between them. How much of a tolerance this would have in age difference could be similar to the age tolerance of examinations (explained in the previous sections 4.4.1, 4.4.2). A difference between

the similarity calculation of the surgery-attribute and the other attributes was that the surgery could be considered a patient specific attribute while the others are directly tied to an examination.

We also made a suggestion about an improvement for matching patients that had undergone surgery, which was to calculate the relevancy of the latest operation of the patient to match other patients with. Our thoughts about this were not based on clinical facts, but rather intuition. We thought that if a patient currently is 16 years old, and went through surgery at an age of 4, then it may not be relevant to match this patient with other patients who went through surgery at 4 years old. This suggestion was not clearly verified by Elisabet, as we did not have enough time to further discuss about it, but the way this was implemented into the system is explained in section 5.2.5.

5 Implementation

By reusing the code of the prototype created in Enbom's work, we could focus on implementing the domain knowledge into the algorithm more easily since the foundation was already done. The application already included the boilerplate code for integrating with a database and also included a graphical user interface to test with. Using the information we gathered from the domain knowledge survey, we implemented the domain knowledge into the algorithm through several steps. These are explained in the following sections.

5.1 Setting up the database

To start implementing, we had to set up a database with the right parameters to work with. We used "MySQL Server" to create a local server, and after signing a non-disclosure agreement, we received an Excel file that contained a snapshot of the CPUP-registry. To skip doing redundant work by processing the 22 000 row large excel file into a database (which Enbom [1] already had done in her work), we received a database dump from Enbom which we imported into a database. In its initial state, the database contained 9 tables and we created an ER diagram of the database at this point to get an overview of its structure. An image of the ER diagram can be seen in figure 3.

Since we had gathered knowledge through our contact with Elisabet about three other attributes that were of interest, we manually added these by importing their respective columns from the Excel file into the database. These attributes were regarding standing aid, which we added into a table called "standing". An updated ER diagram of this can be seen in figure 4.

5.2 Retrieval of similar cases

The source code of the prototype from Enbom's work [1] consisted of a SQL query that was used to fetch information through a local server about cases that were to be compared with the selected patient. After fetching the information, it then looped through each of these cases to run the similarity calculations on each attribute between the patients. This is the part of the algorithm that we decided to put most of our effort on in our work, and will hence be called the "main-loop" in the rest of the report.

In our code, we decided to retrieve the correct attribute-values to match with through a similar SQL query of that which was used in the source code by Enbom, but with updated attributes. This fetch of attribute-values was also decided to be done through our locally connected MySQL server. The query fetched cases that were to be compared with the selected patient (patient c), and since each case represented an examination of a patient, the query retrieved a large amount of rows (about 26 000 rows). This meant that the main loop of the algorithm was very resource-heavy to run. With this in mind, we filtered out a large portion of rows by including our GMFCS matching in the query. Since the GMFCS was to be strictly matched, we could edit the query so that it only retrieved those examinations which consisted of the same GMFCS-level as patient c .

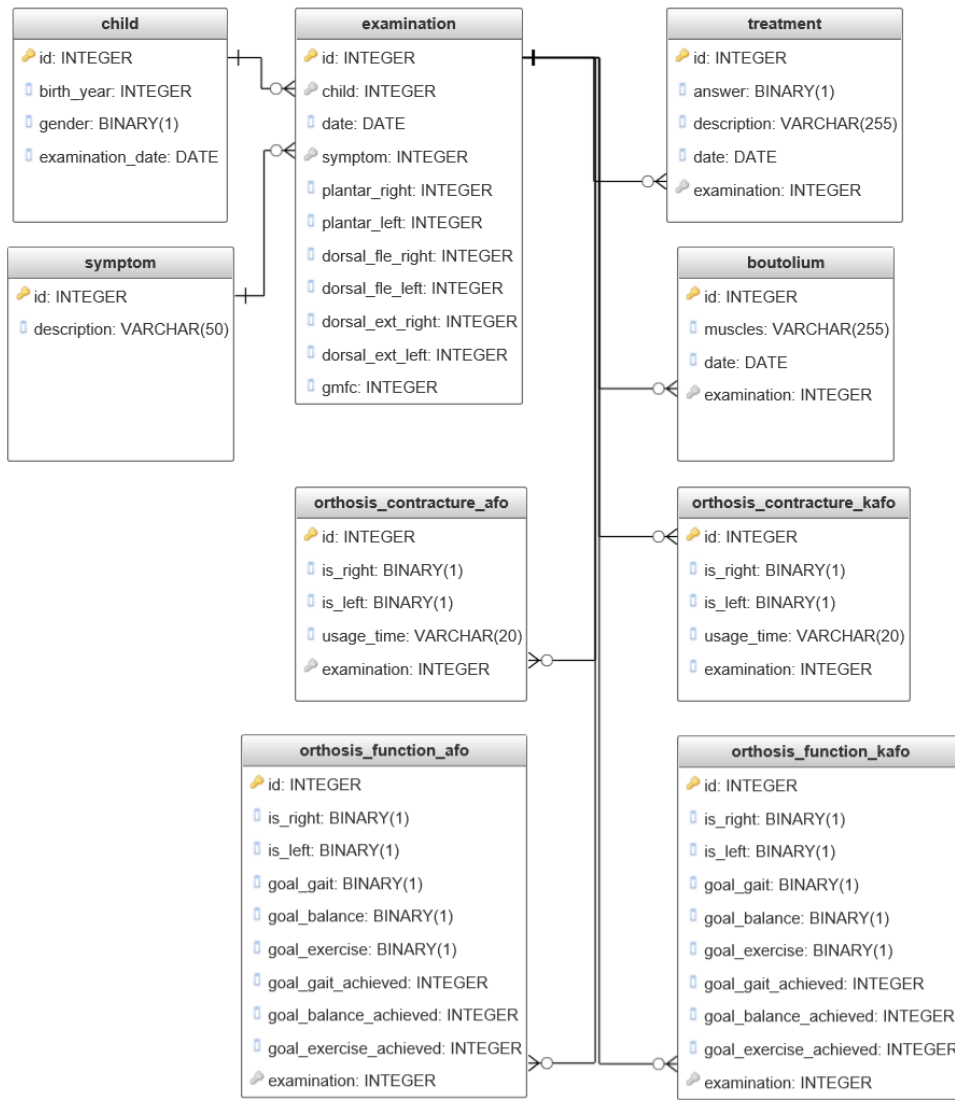


Figure 3: Initial ER-diagram of database created by Enbom [1].

5.2.1 Dorsiflexion similarity

Since dorsiflexion is separated into two attributes, with extended knee and flexed knee, there would naturally had to be two different similarity calculations for them. Since they are similar in that they both measure ankle joint by degrees however, we were not sure how we could calculate the similarity of the two attributes differently. We did not receive any information about treating them differently in their similarity calculations, so we interpreted it as not being too important to use different similarity calculations for them. We thereby decided to treat them similarly, and they will be explained as the same attribute in this section.

As explained in section 4.1, we received information that a difference of 20 degrees between two values of dorsiflexion was significant. This was important information, but we did not receive any more detailed knowledge about other significant differences. We thereby decided to create intervals of differences in degrees based on intuition, where each interval represented a similarity-value. The larger the difference, the less of a similarity value would be given. We also included an additional weight for the similarity-value of dorsiflexion, which was a static value of 0.8 that was to be multiplied with the similarity-value whenever two matched values were not of the same sign. This was an attempt to emphasize the importance of the 0 degree mark, which is also explained

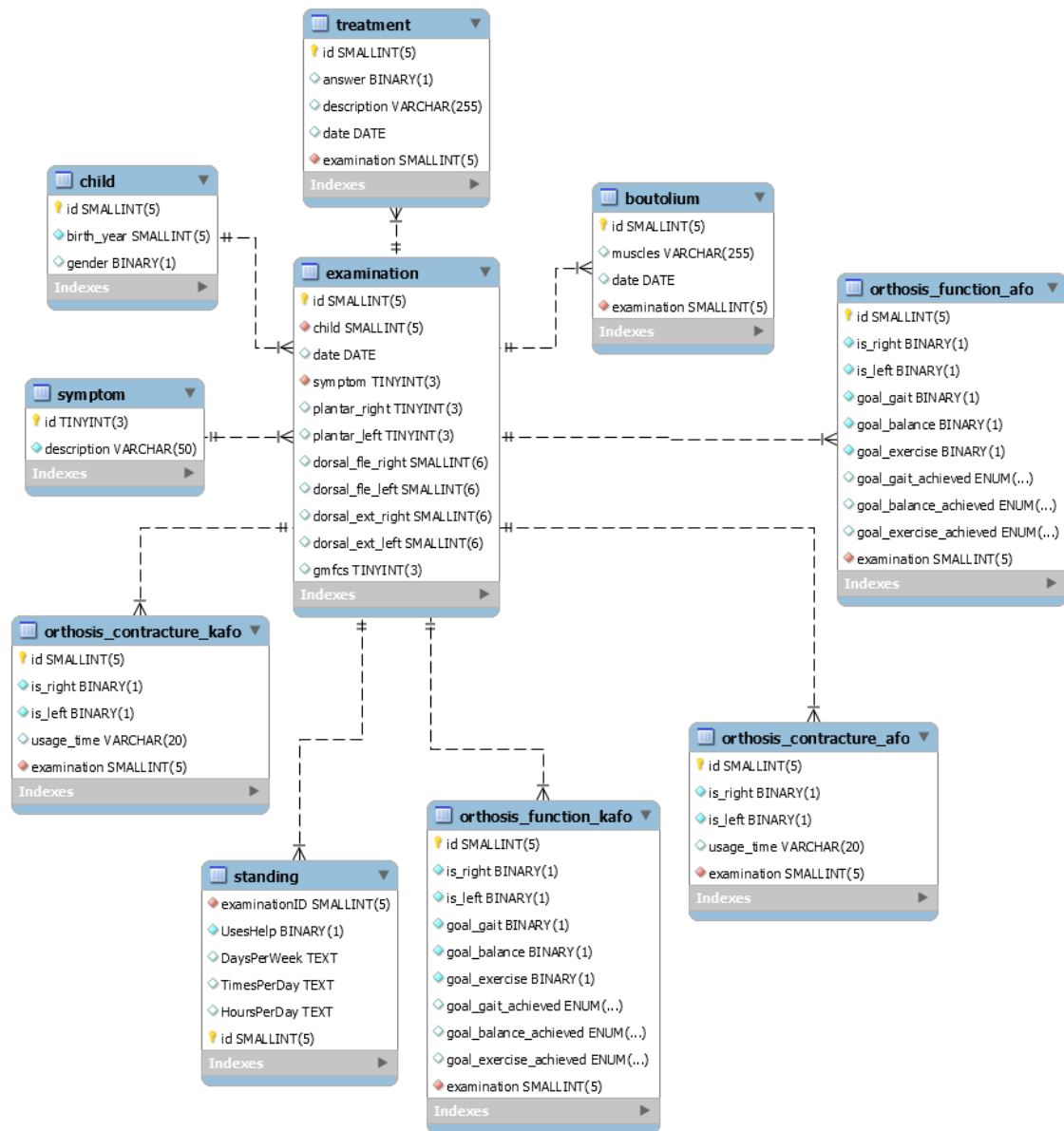


Figure 4: Updated ER diagram, based on Enbom's work, with the "standing" table added.

about in section 4.1. The value of 0.8 was merely based on own intuition.

Due to the importance of dorsiflexion (since it could represent the outcome of treatments, as explained in section 4.1), we tried to set the weight of dorsiflexion accordingly. It was to be of a lower value than age and GMFCS, but higher than the rest of the chosen attributes. The final weight-values we chose can be seen in table 7.

5.2.2 Plantar flexion similarity

As described in section 4.1, Plantar flexion is categorized by five levels, and the suggested way of matching similarity between the levels was to decrease the similarity percentage by 25% for each level that differs. This means that a level 1, for example, is 25% similar to a level 4 and 75% similar to a level 2. This was the way we decided to implement the similarity calculation for the attribute, and we also decided to give it the smallest weight of every attribute so that it would have the least impact on the total similarity between patients. This was done because of the lack of detailed knowledge we had about it in terms of similarity assessment.

5.2.3 Standing aid similarity

When implementing the similarity calculations for the standing aid attributes, we saw the benefit of using the yes/no column for "Uses standing Aid" to allow the program to immediately check if the two other attributes should be compared for similarity. If the answer for "Uses standing Aid" was the same for both patients, 100% similarity was given for that attribute. However, only if the answer was "yes" would the program calculate similarity for the two other attributes "Days per week" and "Hours per day", since a patient with an answer of "no" would not have any entry for the two attributes anyway.

We wanted to simplify the process of calculating the two attributes "Days per week" and "Hours per day", since we thought that they represented almost the same thing but with different units of measurement. We decided to multiply the values of the two attributes to calculate a value that represents the hours per week that a patient uses standing aid, like our suggestion implied in section 4.2. This was to be done for both patients, and the resulting values were to be compared for difference.

Since no detailed domain knowledge was gathered about how to calculate the similarity for the differences between patients regarding standing aid, we used threshold values based on our own intuition to conclude how much of a difference could be tolerated for the attributes to be 100% similar. These can be seen in table 6 under "System Description". We gave more weight to the attribute "Uses standing aid" than the other two, since we thought that it would be more important to match if the patients used standing aid or not to begin with.

5.2.4 Age similarity

The age similarity between two patients is complicated to calculate for several reasons explained under section 4.4. As our suggestion points out in section 4.4.1 and section 4.4.2, we wanted to implement a system that intelligently matches age in a way that would more realistically represent the way a clinician thinks about the similarity between two patients' ages. For example, there could be only a few weeks until a child's birthday, which meant that we had to implement a system that keeps this in mind.

A thing we thought was reasonable to implement was that if the age similarity between the patients has a value of 0, then the matched patient should be disregarded as a candidate for matching. The reasoning behind this was that age has the most impact on the similarity between patients next to GMFCS, and is seen as one of the most important values to be used for similarity matches. It should therefore not be possible to match a 4 year old with a 15 year old for example, no matter how well the other attributes match. This was based on our own reasoning and interpretation of how Elisabet explained the importance of age similarity.

Even with the implementation of dynamic age scaling and age tolerance with the use of graphs and points, which is described in section 6.4, a problem remained with our implementation. To explain it as simple as possible we use an example: An age-group of 10-12 has a tolerance-value of 3, and an age-group of 8-9 has a tolerance-value of 1. A child of age 12 could then be matched with a 9 year old, and not vice versa. When we started implementing this part with age represented by integers only, the solution for the problem was easy: to calculate the supposed minimum tolerated age of an age of 12, we could simply use the maximum tolerated age of its current minimum tolerated age. This would be the maximum tolerated age of 9, which is 10. In code it could look like the following:

Algorithm 1: Simple calculation of a minimum tolerated age by using integers

Data: inputAge = 12, minToleratedAge of inputAge = 9

Result: minToleratedAge of inputAge = 10

```
1 set minToleratedAge of inputAge to (maxTolerance of minToleratedAge) +
   minToleratedAge;
```

When we started to use dynamic tolerance-values however, the results from the example above were not correct. We wanted to find the age of which maximum tolerated age is the same as the input age. Using the same tolerance-values for the two age groups as the previous example: if the input age is 12, we needed to find the age which maximum tolerated age is 12. The age 10 would not be the answer here, since that would have a maximum tolerated age of 13. Instead we would

be looking for a value somewhere between 9 and 10. As it happens, this range of 9-10 is between the two points in the graph prior to the points surrounding the input age. The solution we came up with was to therefore select the previous two points to get a fraction of a curve, which allowed us to run algorithm 3 with the input age as parameter, and re-order equation 3 so that we solved it for *age* instead. This solution only applies to the ages which lowest tolerated age lies within the two previous points of the graph, which we had keep in mind when implementing it.

The first step we took to implement the solution was to rewrite equation 3 so that we could more easily interpret how to write its final form. We wanted to find the age x that, added with its tolerance-value, would give us the specified age:

$$specifiedAge = ((x - lowerX)/distanceX) * distanceY + lowerY + x \quad (1)$$

We could see that if simply re-ordering so that we solve the equation for x , and enter the corresponding values for each parameter, we would get the correct age x . Rewritten so that it was solved for x , the equation looked like the following:

$$x = \frac{distanceY * lowerX + specifiedAge * distanceX - lowerY * distanceX}{distanceY + distanceX} \quad (2)$$

The resulting minimum tolerated age (of the specific ages affected by the problem) was used for the first two points of the similarity-graph of the specified age. This way it created a curve which lowest tolerated age was the age that had a maximum tolerated age equal to the specified age. By utilizing this calculation, there would not be any differences in matching ages of increased or decreased value.

5.2.5 Surgery similarity

When implementing the similarity calculation for surgeries, we decided to use a similar algorithm as when calculating age similarity between patients. The calculation would use the same concept of utilizing 2D-graphs, but instead of using the age of the patients at the time of examination for this calculation, we used the age of the patients at the time of their surgeries. This was not a very apparent approach since surgeries were not directly connected to an examination like the other attributes, but rather the patients as a whole.

As explained in section 4.5, surgeries could imply long term-effects. We therefore decided to make the fall-off value for the graphs of when calculating the similarity values for surgeries a bit higher. We thought that surgeries could still be relevant even though there might be a bit higher difference in age. The final calculation for surgery similarity is shown by pseudo-code in algorithm 5.

5.3 Graphical User Interface

We used the GUI that was provided with the code from Enbom [1], and adapted it to our algorithm to fit our needs. The GUI already provided a way to select values for different attributes of a new patient to be matched. These were for example the age and range of motion for dorsiflexion of the new patient. It is important to state though, that these attributes are different from the ones we developed in terms of similarity assessment. The original GUI for this window is shown in figure 5.

Since we added a couple of new parameters to the algorithm, we added these to this GUI-window. The new parameters were the following:

- GMFCS-level (1-5)
- "Uses standing aid" (yes/no)
- "Uses standing aid: hours per day" (4 alternatives)
- "Uses standing aid: days per week" (4 alternatives)

Figure 5: Original GUI of window where the user can enter new patient parameters, as part of the prototype created in the work by Enbom [1]

Figure 9 shows the updated version of the GUI.

When a new patient was entered or if you selected an existing patient from the database, you could choose to show similar patients, which would run the algorithm and present the results in a list in a new window. The GUI for this window in its original state provided the user with information about the patients' age, range of motion and information about surgeries, boutolium and orthosis. An example of this can be seen in figure 6. Since we had added new parameters to the algorithm, we also added information about these in the GUI so that you could, in a simplified way, see the relevancy of the matched patients. The information we added was the GMFCS-level and the standing aid information about each patient. The updated GUI is shown in figure 10.

There were also a part of the GUI from Enbom's prototype that presented similarity of attributes in the form of graphs. An example of this graph is shown in figure 7, where the x-axis represents difference in degrees, and the y-axis represents similarity. The small circle on the line represents the actual difference between the two patients by its x-value and the given similarity by its y-value. We updated the program with the ability to see the graphs created for the new age similarity-calculations, so that we could get proper feedback about our implementation internally and also show graphically to experts how it works. An example of this graph is shown in figure 12. The x-axis in this graph represents the age (by years, month), and the y-axis the similarity, which is explained further in section 5.2.4. The circle on this graph represents the matched patients age

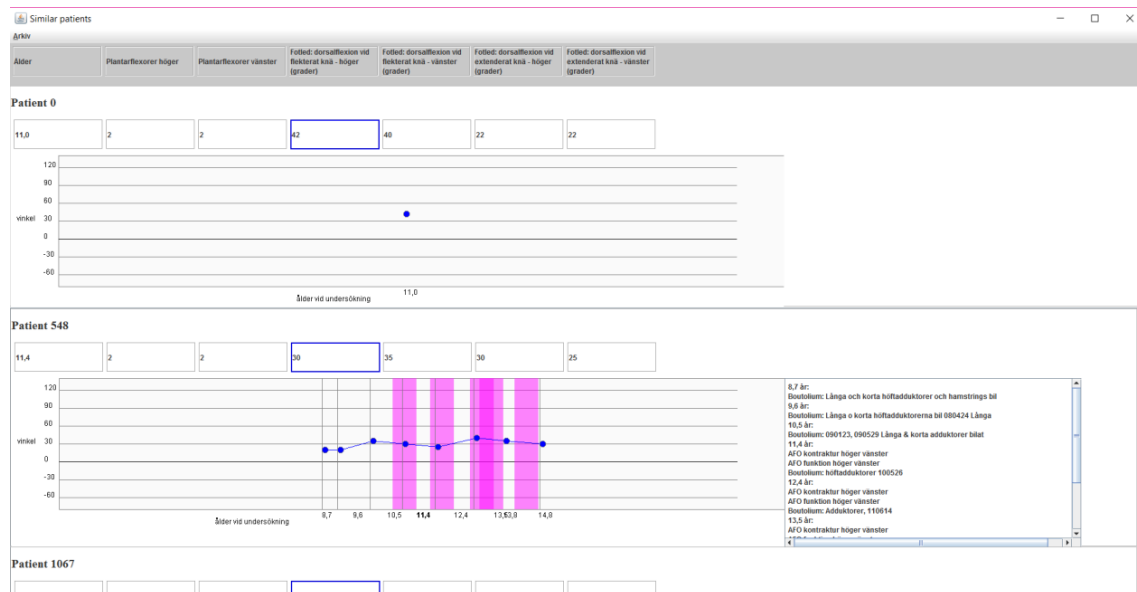


Figure 6: Original GUI of window with results list of matched patients, as part of the prototype created in the work by Enbom [1]

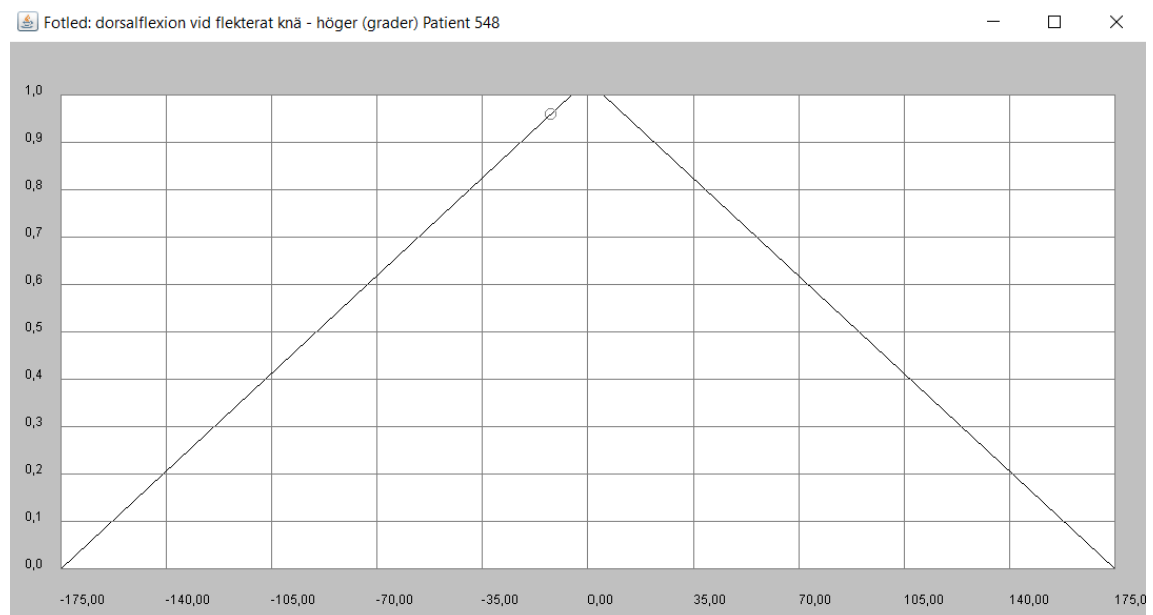


Figure 7: Similarity graph of attribute "Ankle joint: dorsiflexion with flexed knee - right (in degrees)" from Enbom's prototype [1].

by its x-value and the given similarity between the ages by its y-value.

The program included a section where you could see a summation of the similarity-values in the form of a patient similarity table. Using this table, we could easily get feedback about how much of a total similarity-value each patient was given and how much each attributes affected it. Obviously, this table did not include the new attributes, so we decided to update it. The original table can be seen in figure 8 and the updated in figure 13.

Similarities

	ålder	plantar_right	plantar_left	dorsal_fle_right	dorsal_fle_left	dorsal_ext_right	dorsal_ext_left	
Din patients värde	11	2	2	42	40	22	22	
Vikt	0,250	0,125	0,125	0,125	0,125	0,125	0,125	
Patient 548 värde	11,4	2,0	2,0	30,0	35,0	30,0	25,0	
Patient 548 skillnad	0,355	0,000	0,000	-12,000	-5,000	8,000	3,000	
Patient 548 similarity	1,000	1,000	1,000	0,959	1,000	0,983	1,000	
Patient 548 totalt	0,250	0,125	0,125	0,120	0,125	0,123	0,125	0,9928
Patient 1067 värde	11,6	2,0	2,0	40,0	50,0	5,0	15,0	
Patient 1067 skillnad	0,558	0,000	0,000	-2,000	10,000	-17,000	-7,000	
Patient 1067 similarity	1,000	1,000	1,000	1,000	0,970	0,933	0,989	
Patient 1067 totalt	0,250	0,125	0,125	0,125	0,121	0,117	0,124	0,9865
Patient 473 värde	11,0	2,0	2,0	30,0	25,0	20,0	15,0	
Patient 473 skillnad	0,022	0,000	0,000	-12,000	-15,000	-2,000	-7,000	
Patient 473 similarity	1,000	1,000	1,000	0,959	0,939	1,000	0,989	
Patient 473 totalt	0,250	0,125	0,125	0,120	0,117	0,125	0,124	0,9859
Patient 516 värde	11,9	2,0	2,0	20,0	30,0	20,0	15,0	
Patient 516 skillnad	0,876	0,000	0,000	-22,000	-10,000	-2,000	-7,000	
Patient 516 similarity	1,000	1,000	1,000	0,900	0,970	1,000	0,989	
Patient 516 totalt	0,250	0,125	0,125	0,113	0,121	0,125	0,124	0,9823

Figure 8: Original GUI of window with a patient similarity table, as part of the prototype created in the work by Enbom [1]

6 System Description

In the following sections the implemented algorithm, that was developed in attempt to answer the research questions of this thesis, is presented. The attributes used in the algorithm and their respective similarity calculations have been validated by experts in the field of CP.

The algorithm can mainly be explained by its similarity calculations for each attribute between patients. To receive patients to match with however, it needs to receive information about which patient c that should be selected to match with others. This can be done through user input, either by selecting a patient directly from the database or by manually entering values of a new patient. Manual input is done through the GUI presented in figure 9.

After patient c has been selected or created through user input, the algorithm retrieves relevant values to match with through a SQL query to a local MySQL server. Each row fetched by this query represents one examination occasion, or case, of a patient. Each case includes columns of the relevant attributes for the specific examination occasion. The query filtrates the selection of cases by GMFCS-level so that only cases with the same GMFCS-level as patient c are fetched, which makes the table not as large to process.

After fetching all cases, the algorithm enters a loop to process each case for similarity measurement between the case and patient c . The algorithm calculates the similarity between the patients' attributes that are shown in table 1. Each of these similarity-calculations outputs a similarity value for the specific attribute, which is multiplied by a corresponding weight, and then cumulatively added into a total sum. The sum represents the total similarity value between the two patients, and is added into a vector after the similarity calculations have been done for all

The screenshot shows a web-based form titled "Decision support - Din patient". At the top, there is a button "Hämta patient från databas". Below this, the form is organized into several sections:

- Alder:** A horizontal slider ranging from 2 to 20, with a value of 11 selected.
- GMFCS:** A horizontal slider ranging from 1 to 5, with a value of 3 selected.
- Använder ståhjälpmiddel:** Radio buttons for "Ja" (selected) and "Nej".
- Använder ståhjälpmiddel: Antal timmar per dag:** Radio buttons for "<1", "1-2", "3-4", and ">4".
- Använder ståhjälpmiddel: Dagar per vecka:** Radio buttons for "1-2", "3-4", "5-6", and "7".
- Use ankle joint angles:** A checked checkbox.
- Plantarflexorer höger:** A horizontal slider ranging from 0 to 4, with a value of 2 selected.
- Plantarflexorer vänster:** A horizontal slider ranging from 0 to 4, with a value of 2 selected.
- Fotled: dorsalflexion vid flekterat knä - höger (grader):** A horizontal slider ranging from -45 to 125, with a value of 40 selected.
- Fotled: dorsalflexion vid flekterat knä - vänster (grader):** A horizontal slider ranging from -45 to 125, with a value of 40 selected.
- Fotled: dorsalflexion vid extenderat knä - höger (grader):** A horizontal slider ranging from -70 to 110, with a value of 20 selected.
- Fotled: dorsalflexion vid extenderat knä - vänster (grader):** A horizontal slider ranging from -70 to 110, with a value of 20 selected.

At the bottom of the form, there is a button "Visa liknande patienter".

Figure 9: Updated GUI of when entering new patient parameters, based on the prototype created in Enbom's work [1]

attributes. The maximum total similarity between two patients is 1 if all attributes are matched with 100% similarity.

After having calculated the total similarity for all patients, the algorithm selects the 10 closest matches (or 10-nearest neighbours), which is done by sorting the vector of total similarities and selecting the 10 cases with most similarity. These are to be presented by the system.

The similarity-calculation loop and the fundamental parts of the algorithm can more easily be interpreted by the following pseudo code:

Algorithm 2: Main similarity-calculation loop of the algorithm

```

1 collect information about selected patient  $c$ ;
2 retrieve patients data from database where column GMFCS = GMFCS of  $c$ ;
3 while not at end of rows do
4   get next row;
5   set totalSimilarity to 0;
6   calculate age similarity;
7   calculate standing aid similarity;
8   calculate surgery similarity;
9   calculate dorsiflexion/plantarflexion similarity;
10  add results from all calculations to totalSimilarity;
11  add totalSimilarity and patient information to patientVector;
12 end
13 sort patientVector by totalSimilarity;
14 store 10 closest matches;
```

The presented list of 10 closest matches is represented in the GUI to easier grasp the resulting matches. An example of a presented list in the GUI is shown in figure 10.

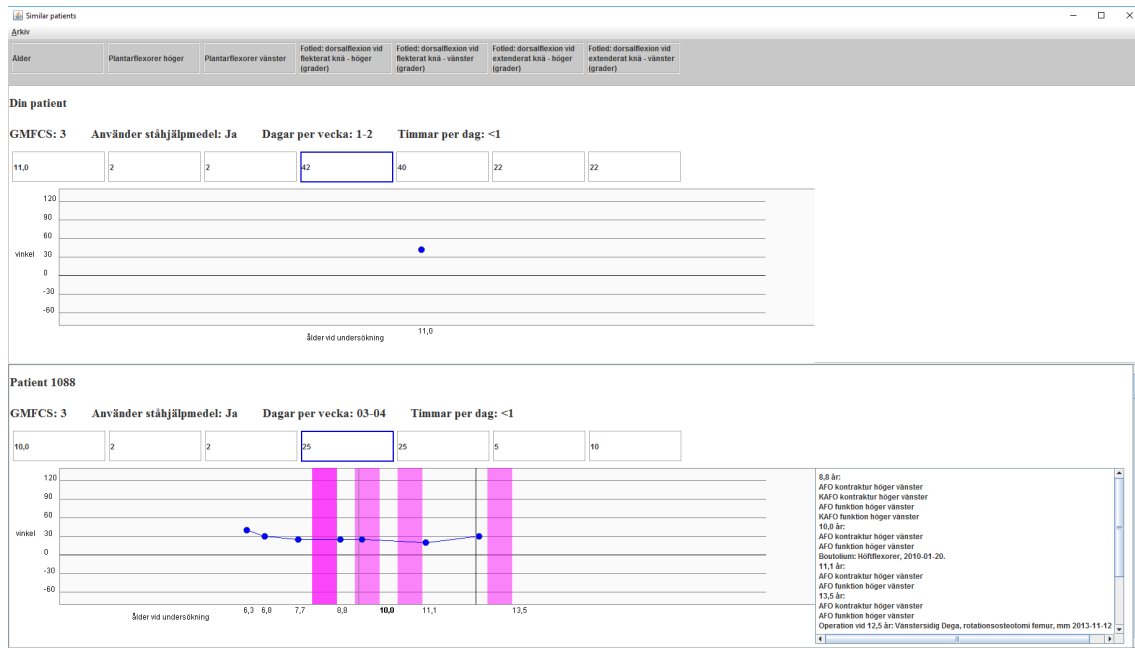


Figure 10: Updated GUI of window with results list of matched patients, based on the prototype created in Enbom's work [1]

The similarity calculation for each attribute differs, and the different similarity calculation are therefore split up and explained in their own section below.

6.1 Dorsiflexion

Dorsiflexion, with both extended and flexed knee for left and right side, is matched based on the amount of difference in degrees there is between the values of each patient. Threshold values for the difference is used to calculate the similarity value. These can be seen in table 4. On top of the calculated similarity value, a penalty value of 0.8 is multiplied if the two values are of opposite sign, since the 0 degree mark is seen as a critical point for dorsiflexion.

Similarity-value	Difference in degrees
1	≤ 10
0.75	10 - 15
0.5	15 - 20
0.25	20- 30
0	>30

Table 4: Threshold values for difference of dorsiflexion (by degrees)

6.2 Plantar flexion

The similarity calculation for plantar flexion is based on the difference between the levels of two patients' plantar flexion. For every step of level difference, we deduct 25% on the similarity value. Table 5 shows the similarity values given depending on the difference in level steps.

Similarity-value	Difference in level steps
1	≤ 0
0.75	1
0.5	2
0.25	3
0	>4

Table 5: Threshold values for difference of plantar flexion (by level steps)

6.3 Standing aid

The value for the attribute "Uses standing aid" is represented as a binary value, or "yes/no", and the algorithm checks if this value matches for the two patients. If it does, a similarity of 100% is concluded. The algorithm will also calculate the similarity of the two other attributes "Days per week" and "Hours per day". The values from these two attributes are multiplied to get a value that represents hours per week that a patient uses standing aid. This is done for both patients, and the resulting values are then compared for difference. The lower the difference is, the more similarity-value is given. Threshold values for this was used and can be seen in table 6 below.

Similarity-value	Difference in hours
1	≤ 1
0.75	1 - 3
0.5	3 - 5
0.2	5 - 10
0	≥ 10

Table 6: Threshold values for difference in standing aid usage time (hours per week)

6.4 Age similarity

The algorithm calculates the similarity between two patients' age by using two separate 2D-graphs. These graphs are created by specifying points by positions of x and y-values. Lines are then drawn between each point to create the curve of the graph. The first graph represents the similarity-value by its vertical axis, and the age of a patient by its horizontal axis. This graph is used to find what the similarity-value of an age should be, based on an inputted age (x-value). The curve of the graph is based on the tolerated age difference that the age of patient *c* has.

To calculate the tolerance of a specific age, the second graph is used. Its vertical axis represents tolerance-value, and the horizontal axis represents the age of a patient. In this graph, points are created based on the values from table 3, which means the curve of this graph will remain the same for every age. The created graph is shown in figure 11. The input for the graph is the age of patient c , and the output is a tolerance-value which is used for the points of the similarity graph.

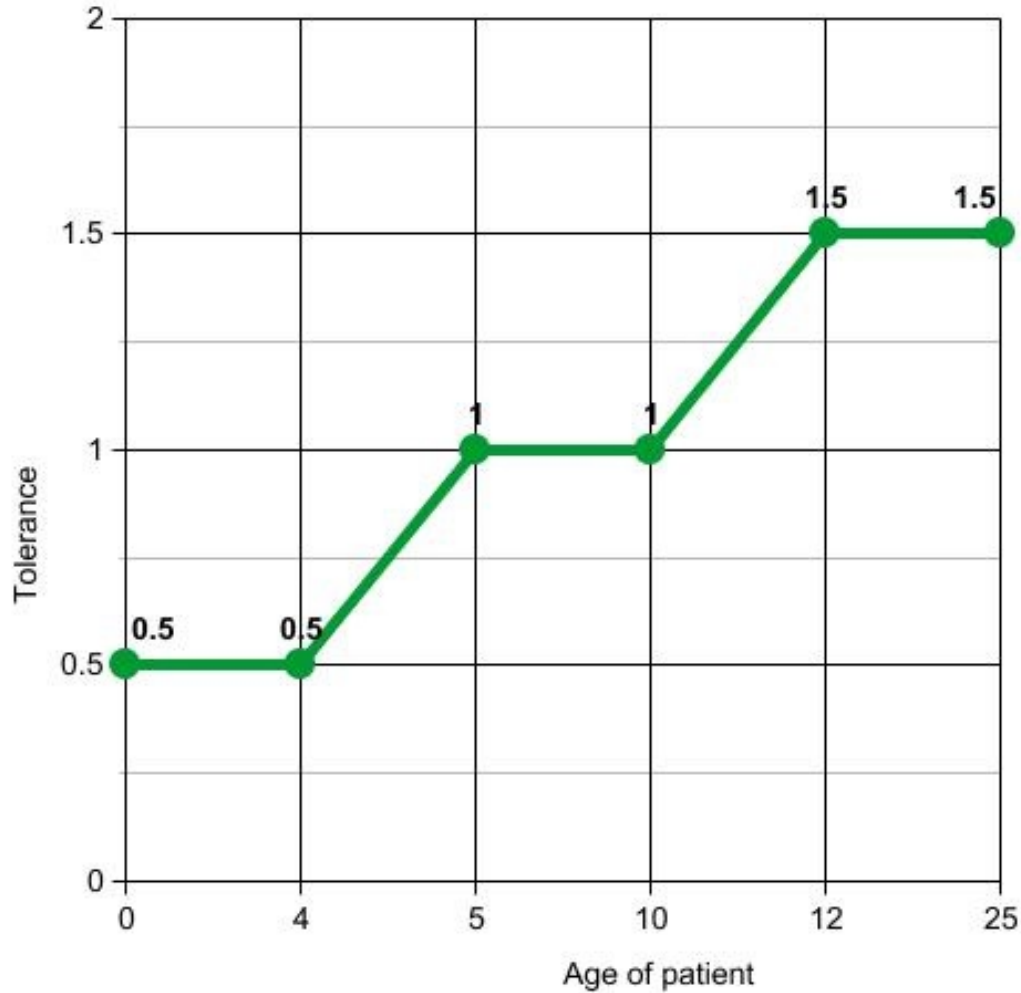


Figure 11: Graph used to calculate tolerance-values for specific ages.

Since each age will be given an own tolerance value, points of the similarity graph are dynamically created with an x and y value, based on the age and tolerance of patient c . The points at the beginning and end of the similarity graph are also given an extra x -value so that the curve will be using some fall-off that smooths the transition from the maximum or minimum tolerated age from similarity 1 to 0. This value is set to 2 when calculating age similarity. This means that an age will still be able to be matched with an age outside of its tolerance-range, only with less similarity-value given. Graphical representations of the similarity graph can be seen at run-time through the GUI to get direct feedback of how it looks and where in the curve that the matched age were positioned. An example of a similarity graph presented in the GUI is illustrated in figure 12.

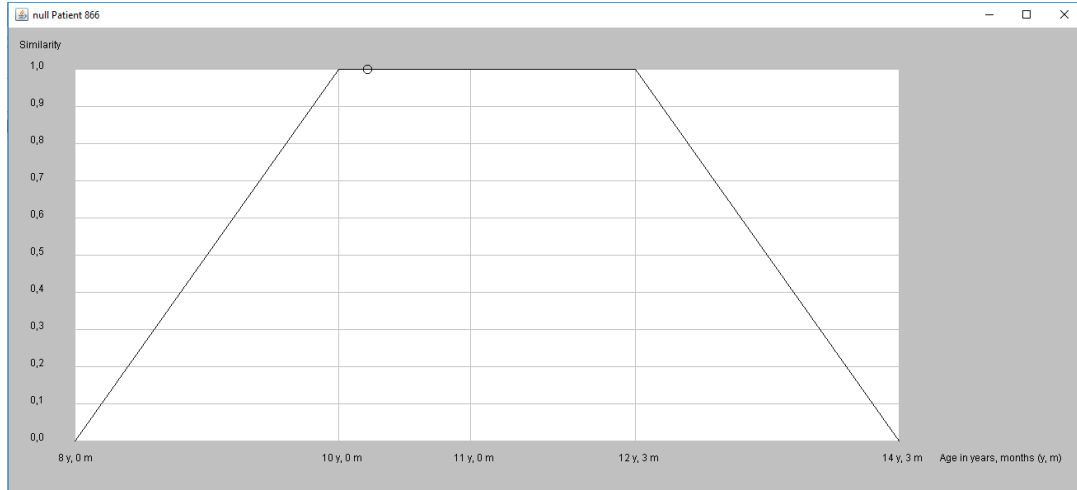


Figure 12: Window of graph used to calculate the similarity-value between two ages, as shown in the GUI. GUI is based on the prototype created in Enbom's work [1]. Curve of the graph is based on tolerance values by an age of 11. An age of 10 years and 2 months used as input, represented by the circle.

The way the algorithm calculates the output-value of the graphs is by selecting the two points that surrounds the inputted age, based on their x-values. An example would be, if the inputted age is 9, to select the point with an x-value of 8 as the lower point, and the point with an x-value of 10 as the higher, assuming there are no other points in between those two. The output-value is then be calculated by using the following equation:

$$output = (((age - lowerX)/distanceX) * distanceY) + lowerY \quad (3)$$

where *output* is the output-value for the entered *age*, *age* is the entered age, *lowerX* is lower point's x-value, *lowerY* is lower point's y-value, *distanceX* is *higherX* - *lowerX*, *distanceY* is *higherY* - *lowerY*.

By using this equation, the calculation of the tolerance-value can be explained with the following pseudo-code:

Algorithm 3: Algorithm for calculating tolerance-value of an age

```

1 add points to tolerance-graph based on table;
2 for  $i = 0$  to  $points.size$  step 1 do
3   if x-value of  $point[i]$  is greater or equal to age then
4     set point[i] as higher point and point[i-1] as lower point;
5     exit loop;
6   end
7 end
8 if  $i$  is equal to 0 then
9   set tolerance to y-value of first point
10 else
11   calculate maximum age difference using equation 3;
12   set tolerance to resulting maximum age difference;
13 end

```

After calculating the tolerance-value, it is used to create the points of the similarity graph in order to calculate the final similarity. If an age has a minimum tolerated age which lays between the previous two points of the tolerance-graph then the lower points of the similarity graph are adjusted to consider dynamic tolerances of other ages. This is a result of the problem explained in section 5.2.4 about matching ages with increased or decreased matched values. The way the algorithm knows this before calculating it is by using the values of the tolerance table 3. The lower points of the similarity graph are adjusted by calculating the supposed minimum tolerated age using

algorithm 3, but with the use of equation 2. Other than that, the calculation of the similarity-value is similar to algorithm 3. The following psudeo-code explains the similarity calculation using the similarity graph:

Algorithm 4: Algorithm for calculating similarity-value between two ages

```

1 calculate tolerance of currentPatientAge using algorithm 3;
2 calculate estimated lowest age of currentPatientAge using tolerance;
3 if estimated lowest age is between previous two points of tolerance-graph then
4   calculate minimumToleratedAge of currentPatientAge using algorithm 3 with equation 2;
5   add point with x-value of (minimumToleratedAge - fallOff), and y-value of 0;
6   add point with x-value of minimumToleratedAge, and y-value of 1;
7 else
8   add point with x-value of (currentPatientAge - tolerance - fallOff), and y-value of 0;
9   add point with x-value of (currentPatientAge - tolerance), and y-value of 1;
10 end
11 add point with x-value of (currentPatientAge + tolerance), and y-value of 1;
12 add point with x-value of (currentPatientAge + tolerance + fallOff), and y-value of 0;
13 for  $i = 0$  to  $points.size$  step 1 do
14   if x-value of point[i] is greater of equal to other patient's age then
15     set point[i] as higher point and point[i-1] as lower point;
16   end
17 end
18 if  $i$  is equal to 0 then
19   set similarity to y-value of first point
20 else
21   calculate similarity using equation 3, where age corresponds to other patient's age;
22 end

```

6.5 Surgery similarity

The algorithm fetches information from the database about every surgery that the patients has undergone. It then compares the age of a fetched patient when they went through their surgeries with the age of the selected patient c 's age when he/she went through the most recent surgery. This is done in a similar method as in the previous section with graphs and curves. The fall-off values for the points used with surgeries is set to 4 though to be able to match less strictly. If a patient has undergone several surgeries, they are all compared with the most recent surgery done by patient c . Only the surgery that receives the most similarity (closest match by age) is selected to be used as output similarity.

After the similarity has been calculated, its value is multiplied by a weight like all other attributes. It does however multiply by another value as well, which is a relevancy-value of the surgery made by patient c . The calculation of this value is identical to the one between two different patients' surgery age. The two input ages for the calculation of relevancy are the current age of patient c and the age of patient c at the time of the most recent surgery. This means that the algorithm also considers how much of a similarity-value should be provided to the total similarity value between patients based on how relevant the latest surgery of patient c is in conjunction with its current age.

The following code explains the process of calculating a similarity-value between surgeries:

Algorithm 5: Algorithm for calculating similarity-value between two patients' surgeries

```

1 find latest surgery made by current patient;
2 calculate patient age at the time of latest surgery;
3 set surgeryAgeCurrent as resulting age;
4 set fallOff-value of points for curve in algorithm 4 to 4;
5 calculate similarity between current age and surgeryAgeCurrent using algorithm 4;
6 set relevancy as the resulting similarity;
7 retrieve surgery data of other patient;
8 set resultingRows as rows from retrieval;
9 initialize ageClosestDiff;
10 initialize ageClosest;
11 while resultingRows is not at end of rows do
12   get next row of resultingRows;
13   set surgery as next row;
14   set surgeryAgeOther as calculated age of other patient at surgery date;
15   set ageDiff as surgeryAgeCurrent - surgeryAgeOther;
16   if ageDiff is less than ageClosestDiff then
17     set ageClosestDiff as value of ageDiff;
18     set ageClosest as value of surgeryAgeOther;
19   end
20 end
21 calculate similarity between surgeryAgeCurrent and surgeryAgeOther using algorithm 4;
22 set similaritySurgeries as resulting similarity;
23 set totalSurgeriesSimilarity as similaritySurgeries * relevancy;

```

6.6 Patient similarity

Each patient match is given a total similarity value after calculating the similarity-values of every attribute, and a part of the GUI shows a summary of how much each attribute contributed to the total similarity-value. It also shows how much of a similarity-value was calculated for each attribute and their respective weights. An example of this in the GUI is shown in figure 13.

The weight for each attribute is set according to our own intuition, since there were no specific weight-values to be established through the domain knowledge survey that was conducted. Guidelines were given through the domain knowledge study however, which were in the form of relations between the weights and how much they should approximately differ from each other. Based on these, we set the weights to our best ability. The final weights that were set to each attribute can be seen in table 7 below. The weights can however be adjusted by clinicians through a window in the GUI, as shown in figure 14.

Attribute	Weight
Age	0.4
Uses standing aid	0.15
Uses standing aid: hours per week	0.05
Dorsiflexion (extended knee) right/left	0.05
Dorsiflexion (flexed knee) right/left	0.05
Plantarflexion right/left	0.025
Surgery	0.15

Table 7: Weights for the attributes that were used

Age is the factor with highest amount of impact on the similarity between patients, which can be seen in table 7. A filter is used after calculating the age similarity which removes all patients with an age similarity of 0% as candidates for matching. This was to eliminate the risk of matching

two patients that have other attributes similarly matched but also a very large age difference.

	ålder	Help with standing	plantar_right	plantar_left	dorsal_fle_right	dorsal_fle_left	dorsal_ext_right	dorsal_ext_left	Total similarity
Din patients värde	11	-	2	2	42	40	22	22	
Vikt	0,400	0,200	0,025	0,025	0,050	0,050	0,050	0,050	
Patient 505 värde	12,0	0,0	0,0	0,0	35,0	35,0	20,0	20,0	
Patient 505 skillnad	1,015	0,000	-2,000	-2,000	-7,000	-5,000	-2,000	-2,000	
Patient 505 simila...	0,800	0,200	0,500	0,500	1,000	1,000	1,000	1,000	
Patient 505 totalt	0,400	0,200	0,013	0,013	0,050	0,050	0,050	0,050	0,8250
Patient 627 värde	11,2	0,0	0,0	0,0	45,0	45,0	30,0	30,0	
Patient 627 skillnad	0,153	0,000	-2,000	-2,000	3,000	5,000	8,000	8,000	
Patient 627 simila...	0,800	0,188	0,500	0,500	1,000	1,000	1,000	1,000	
Patient 627 totalt	0,400	0,188	0,013	0,013	0,050	0,050	0,050	0,050	0,8125
Patient 665 värde	12,3	0,0	1,0	2,0	30,0	30,0	20,0	20,0	
Patient 665 skillnad	1,340	0,000	-1,000	0,000	-12,000	-10,000	-2,000	-2,000	
Patient 665 simila...	0,764	0,188	0,750	1,000	0,750	1,000	1,000	1,000	
Patient 665 totalt	0,382	0,188	0,019	0,025	0,038	0,050	0,050	0,050	0,8007
Patient 527 värde	9,9	0,0	0,0	0,0	40,0	40,0	30,0	30,0	
Patient 527 skillnad	-1,059	0,000	-2,000	-2,000	-2,000	0,000	8,000	8,000	
Patient 527 simila...	0,776	0,188	0,500	0,500	1,000	1,000	1,000	1,000	
Patient 527 totalt	0,388	0,188	0,013	0,013	0,050	0,050	0,050	0,050	0,8007

Figure 13: Updated GUI of patient similarity table, based on the prototype created in Enbom's work [1]

Adjust weights

Alder:

Använder ståhjälpmedel:

Ståhjälpmedel - Timmar per vecka:

Dorsalflexion (extenderat knä) höger/vänster:

Dorsalflexion (flekterat knä) höger/vänster:

Plantar flexion höger/vänster:

Operation:

Spara

Figure 14: Window for adjusting weights through the GUI

7 Discussion

Relevance is the key word when matching similar patients, but what is considered relevant can vary depending on who is asked. What a clinician finds relevant may not be of interest for a physiotherapist and vice versa, so defining what is relevant was rather tricky. The interview with Elisabet gave some clarity in the matter and there were two variables that, according to her, were relevant for every field concerning CP. These two variables were age and GMFCS. We were told to strictly match each GMFCS level independently, but this might not be realistic since there may be factors of GMFCS that may not allow such a filter to be fitting. The age similarity calculation, on the other hand, was implemented intelligently and dynamically so that the age similarity would be as realistic as possible. We believe that having the algorithm intelligently calculate similarities between attributes makes it adapt well for many cases. We think that this increases the longevity of the algorithm since it will be able to handle most cases in the future, even if new correlations are found that affects the relevancy of attributes.

Self-sustainability could also have great impact on the longevity of the CDSS, and it could be achieved by having the algorithm directly import patients from the CPUP database without the meddling of a clinician. It can be further implied that implementing a continuously updating system is of great interest, but due to the risk of exposure for sensitive information, the access needed to gather data directly from an updating database might be hard to get. We think that these future type of improvements are necessary for the CDSS to be viable for real practise, and that without further development the system will not be compelling enough for clinicians to use in their day to day work.

7.1 Significance

The results show how an algorithm can be improved using domain knowledge combined with temporal aspects to find similar cases of patients diagnosed with CP. Determining the significance of the results is depending on the perspective and how large of a scope that is set. Improving a matching algorithm for the purpose of helping clinicians when diagnosing patients with CP may seem rather insignificant when compared to solving world hunger or curing cancer. However, when it comes to enabling clinicians and lead them to better informed conclusions, then the results could potentially help improving the lives of children with CP and that is significant enough to make this type of project worth investigating.

No apparent consequences can be seen from our perspective and we don't believe that this work has big enough impact to create consequences of significance.

7.2 Previous work

By reusing code from the prototype by Enbom [1] while mixing it with our own programming style, it became quite tedious to properly interpret parts of the code after several sessions of programming. This was probably the reason why we stumbled over a couple of bugs, one of which increased the run time of our matching function from 2 seconds to approximately 40 seconds, which was unacceptable. By finding these bugs and fixing them, we could reach 2 second run times again. By doing this extra work we wasted more time than we would have wanted to though, and we would probably have been better off starting building up a code structure from scratch. Since we did not do this, it may also prove to be a large issue for future development in case a third party is to develop the algorithm further. Then they would have to understand the mixed code made by Enbom [1] and us.

7.3 Goals

The aim of this thesis was to explore the possibilities of domain knowledge and temporal aspects in similarity assessments between patients with CP. The purpose was also to investigate how, through the use of domain knowledge and temporal aspects, matching of similar patients with CP could be done through CPUP data in a similar way that experts would have done. In the following sections, we explain how the research questions described in section 1.3 was answered, and thereby how the purpose of this thesis was achieved.

7.3.1 Addressing temporal aspects

As explained in section 2.4.3, there could be many variables identified as temporal aspects, but finding these in the CPUP-registry involves quite extensive work if one does not know exactly what to search for. One type of temporal aspect that we could identify from the CPUP-registry, together with Elisabet, was surgery. There were common enough occurrences of surgeries in the database to be able to use them as similarity assessment for the algorithm, and uncommon enough occurrences for them to be considered temporal aspects. In this work, we addressed surgeries as a single attribute even though there are several different types of surgeries. According to Elisabet, this type of similarity assessment should be acceptable due to the fact that every surgery is similar in the way that they have a drastic and often long lasting effect, which is what makes the aspect so interesting to match with.

To some extent, temporal aspects could generally be addressed by matching patients with similar temporal aspects. This way, clinicians would be provided with further useful information about significant events of patients, provided that the temporal aspects are relevant originally. However, temporal aspects may not always be obviously matched together, since there may be other variables affecting the relevancy of them.

7.3.2 Combining temporal aspects and domain knowledge

The way we could combine temporal aspects in the form of surgeries with domain knowledge was to treat temporal aspects in a similar manner as any other attribute. Each attribute had an own similarity calculation and so could surgeries. But a difference between the domain knowledge attributes and surgery was that surgeries depended on another factor for relevancy, which was the age of the patients at the time of surgery. Another difference was that when calculating the similarity for the different attributes, their values corresponded to those of the examination occurrence in question, while surgeries were occurrences themselves. Surgeries could thereby be considered as patient-specific attributes, while the other attributes were directly tied to an examination. We had to take this in consideration in the similarity calculation, but because of the dependency it made the attribute not as apparent as the other attributes. By adding the similarity value for each attribute, a total sum of similarity between patients was calculated, and through adding the additional similarity of surgeries as well, the algorithm could distinguish the more unique cases with surgeries.

Each similarity value is multiplied by a corresponding weight to calculate the amount of impact the attribute in question should have on the total similarity value between two patients. When matching patients, we search for the k patients with the highest total similarity value. If k would be set to a small number, then each matched patient becomes more important to provide accurate and relevant information, which would mean that each weight needs careful fine tuning. If temporal aspects are included in the matching, then matches involving temporal aspects will likely dominate the small list of matched patients, which may or may not be what a clinician is looking for. With this in mind, we think that utilizing weights is a powerful tool for such scenarios thanks to the flexibility of tuning them to adapt the search for different purposes. All one would have to do is to change the weight for the attribute that should have more or less impact on the total similarity value between two patients. If there are few weights and attributes, this sort of fine tuning is easy to do. If the weights and attributes become many though, then changing the weight of one attribute could have a bigger impact than anticipated, since there are many other weights that then more easily could be overruled.

7.3.3 Improvement of identification using domain knowledge

The algorithm improves identification of similar patients with CP by utilizing domain knowledge in similarity calculations. Since the most important factors of similarity assessment between patients, for clinicians, are age and GMFCS, we can conclude that resulting matches are clearly improved in relevancy by running the algorithm with different selected patients.

The intelligent part of the algorithm, which had most impact on the improvement of the identification of similar patients, resides mostly in the age similarity calculation. Creating an intelligent similarity function for age was an attempt to make the algorithm identify similar ages in a way

that more likely resembles a clinician’s idea of similarity between patients’ ages. Through dynamic scaling and tolerance-values, the algorithm intelligently create transitions between significant age-groups, specified in table 3, so that no strict thresholds are used as similarity measurement. We think that this part of the algorithm, together with the filtration of GMFCS-levels, were two large steps in improving similarity matching between patients with CP, since they seem to have most influence on which treatments that should be considered or not.

If the algorithm were to be used in a CDSS by clinicians in its current state (without considering data limitations), we think it could be of good use. Though, we also think that it could be improved by further expanding on included attributes, hence being able to more adaptive suit the needs of clinicians and what information they would like to find about a patient.

7.4 Justification

An evaluation of the implemented system was done through collaboration with experts in the field. By matching similar patients of two different patients from the database, the results was sent to two experts for verification. The two chosen patients to match other patients with differed by GMFCS level and age, so that the two most important factors would provide different results to verify. From the resulting 10 matches for both patients, the three closest of them were used for verification. This was to eliminate the risk of overwhelming the experts with information. The selected patients and their corresponding three matches were presented to Elisabet and Gunnar Hägglund (Adjunct professor at Department of Orthopaedics at Lund university), and they both verified the results.

Elisabet gave directions for most of the similarity calculations and provided details about how a clinician in general sees the similarity between patients. Based on the knowledge she has in the area, the information she provided and the evaluation done, we think that the algorithm should be justified to some extent.

Apart from the conducted evaluation, we did ongoing internal testing of the algorithm by ourselves throughout the work to ensure that the most important attributes were similar between patients. We also cleaned up any bugs found to our best extent during the testing, since there were some complications when merging with the code from Enbom’s prototype [1]. We think that these procedures ensures a more stable algorithm which will be more easily maneuvered if it is to be used in future development.

7.5 Limitations

As stated in section 1.4.1, the supply and access of data was limited and this could potentially mean that the choice of AI-technique would have been different if we had access to more data. AI-techniques such as GA have potential in this sort of algorithm. However, we believe that the benefit of examining each case separately in case-based reasoning outweighs the benefits of higher efficiency gained from GA. The reasoning behind this is that since the algorithm is searching for relevant information rather than solutions, efficiency is not as prioritized as the ability to handle cases separately. Time limitations made the scope for the project small and only the most important factors could be our focus regarding domain knowledge and temporal aspects. With more time and a larger scope, more work targeting these areas could help improve the algorithm and provide even better results.

Among the chosen attributes in this thesis, some are developed and validated in greater detail than others. Due to limitations in contact and communication with experts, which meant more limited domain knowledge, we put more effort on similarity assessments of the most important attributes such as age and GMFCS. Other attributes such as plantar flexion and dorsiflexion could not be defined with as much precision regarding weights and how to calculate the differences. These were therefore given weights based mostly on our own intuition. To combat the problem of limited domain knowledge in this project, we could have included more experts and let them evaluate a few attributes each. This would have saved time and potentially lead to even better results.

The evaluation of the implemented system was not very extensive. We could not expand this part further due to time limitations and experts tight schedule. The scope of this project was relatively small though, and if the scope of the project would have been larger, the evaluation part

would probably have been more emphasized. However, we think that the conducted evaluation verifies that an algorithm that matches similar patients with CP could be worth to expand further in conjunction with research of a CDSS for CP, and that domain knowledge and temporal aspects can beneficially be utilized in this context.

7.6 The bigger picture

The algorithm we implemented can have appliances for software developers working with CDSSs. They can potentially develop this project further or see this work as related to theirs. By seeing the benefits of CBR, one could adapt this work to its area or use domain knowledge in conjunction with other AI-techniques. The algorithm is, however, very specific to this project and hard to generalize for other purposes. What could be reused from our perspective is the usage of domain knowledge and temporal aspects. We think that the appliances for domain knowledge and combining it with temporal aspects are plentiful, especially in the medical field. This is since research regarding domain knowledge is mainly residing in the medical field, which means it's easiest applicable there. However, that doesn't mean the appliances of domain knowledge and temporal aspects is strictly limited to be used for medical purposes.

The same way we gathered domain knowledge, we see potential in gathering domain knowledge in areas where relevant information in a large amount of data can not be found by calculating statistical relations. As explained in section 3.4, finding connections in data based on statistical calculations is getting more common, but is not something that can be done when there is no clear answer regarding, for example, what treatment a patient should get. We also see potential in gathering information about temporal aspects in areas where there exists temporal data. Uncommon occurrences in the data can prove to be relevant in many different scenarios since it could provide information that may seem out of place or important. As stated in section 2.4.3, there are tools for extracting patterns in temporal data which means that identification of relevant temporal aspects has potential to become easier.

8 Conclusions

This thesis explored the possibilities of domain knowledge and temporal aspects in similarity assessments between patients with CP. The work was structured by three main parts: gathering of domain knowledge, implementation of a matching algorithm and validation of attributes used in the algorithm with experts. The amount of related work is limited and only the work made by Enbom [1] can be seen as strictly related. By expanding the view of what can be considered related work, there is research conducted in the fields of CP, CDSS, Domain knowledge, temporal aspects and CBR that was of interest for this work. We investigated CDSSs targeting other medical areas such as diabetes, and these systems experimented with fuzzy knowledge and other theories when dealing with domain knowledge.

Instead of using experimental theories, we chose to focus on facts provided by experts in the area when implementing domain knowledge into the algorithm. This was done through interviews and other means of communication. The data regarding patients was retrieved from the CPUP database, which is specific to patients with CP, and the amount of data we accessed from the database was not large enough to use AI-techniques such as GA.

Through an investigation aimed to answer research question 2 regarding temporal aspects, we found surgeries as a fitting attribute, but other temporal aspects could have been useful in a larger scope for the project. By treating surgeries as an attribute like the domain knowledge based attributes, but with respect to other variables affecting it, we could combine temporal aspects with domain knowledge and thus answer research question 3. Due to limitations in access of data, the algorithm shouldn't be applied to real practise, but with changes such as direct access to the CPUP database or having clinicians supply the system with new information it could. Other limitations such as scope and time lead to only a handful of values being extracted from domain knowledge and temporal aspects.

Implementing an algorithm for matching similar patients using the gathered knowledge was done through weights and intelligent similarity calculations for different attributes. Through the

implemented algorithm, which similarity calculations are based on expert verification, and that accurately matches similar patients according to the domain knowledge gathered, we answered research question 1.

The resulting implemented algorithm is presented through a prototype of a CDSS which allows clinicians to manually enter specific characteristics for a new patient or select an existing patient from the CPUP database; perform a similarity search using the selected patient, and be presented with the matched similar patients. In the current state of the system, the data is not continuously updated, and in order for this work to stay relevant it would require either updates from clinicians or the ability to fetch data directly from the CPUP database. Usage of the CDSS could help clinicians make better informed decisions, which leads to improved health care for children and patients with CP. This is ultimately why this research is important.

The algorithm was evaluated by a small amount of experts and it was deemed as good. To ensure a good accuracy of the algorithm so that it could be useful for many clinicians, a more extensive evaluation would be required. The conducted evaluation in this work can, however, be sufficient as a proof of concept for matching similar patients with CP using domain knowledge and temporal aspects.

9 Future work

A self-sustaining system with the possibility to fetch data directly from the CPUP database where patients are continuously updated, would be highly interesting for stakeholders using this type of system. It would also be interesting to make the system adapt to new information, so when critical outcomes from examinations is inserted to the system, the system would change its matching accordingly. This would help avoiding poorly matched patients due to old information, and increase the potential of the CDSS.

In the current state of the medical field there are individual CDSSs for almost every sickness or disorder that have active research for it. If there was a possibility to unify several or all CDSSs to one, it would be interesting for clinicians in terms of simplifying the learning process and increase the usability of a CDSS. The idea is that clinician should be able to focus on the medical part of their job rather than learning how several CDSSs works. Many within the medical field could profit from it, and it might lead to a better quality of life for clinicians working with CDSSs, since they could then be more focused on what they were intended to do.

Although the addressing of temporal aspects in this thesis proved to be functional and acceptable, it should probably be expanded in further development to utilize the full potential that could be achieved with temporal aspects. Instead of limiting temporal aspects to surgery and generalizing surgeries to one variable, one could expand both the amount of temporal aspects included and match each surgery by its type.

Puberty is a phase almost every human past 20 years old have gone through. During this period it is common knowledge that hormones affect the body in many ways and investigating how this applies to CP may lead to interesting results. A 15 year old patient that has not entered puberty may be more similar to a 12 year old patient that has not entered puberty, than to a 17 year old patient that has entered puberty.

In a similar manner that standing aid was implemented, the usage of orthoses or corsets could be taken in consideration for matching similarity, and provide additional relevant information for clinicians.

Another potential for future work is to look at the history of a patient and see, for example, how many patients were similar at an age of 7 years, and how many are similar now. This could help with displaying the outcome and development of patients with their symptoms.

Graphs could be useful to help smooth the transitions in similarity calculations for every attribute, and not just for age. Rather than using threshold-values with strictly drawn lines, this could lead to more realistic and accurate similarity calculations for many attributes, which could potentially increase the relevancy of matched patients further.

References

- [1] A. Enbom, “Clinical decision support system for cerebral palsy,” Master’s thesis, Mälardalen University, 2016.
- [2] R. Kumar and A. Sharan, “Personalized web search using browsing history and domain knowledge,” in *2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, Feb 2014, pp. 493–497.
- [3] M. M. Richter and R. O. Weber, *Basic Similarity Topics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 113–147. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40167-1_6
- [4] A. J. Aljaaf, D. Al-Jumeily, A. J. Hussain, P. Fergus, M. Al-Jumaily, and K. Abdel-Aziz, “Toward an optimal use of artificial intelligence techniques within a clinical decision support system,” in *2015 Science and Information Conference (SAI)*, July 2015, pp. 548–554.
- [5] M. U. Ahmed, S. Begum, and P. Funk, “Case studies on the clinical applications using case-based reasoning,” in *2012 Federated Conference on Computer Science and Information Systems (FedCSIS)*, Sept 2012, pp. 3–10.
- [6] E. S. Berner, *Clinical decision support systems*. Springer, 2007.
- [7] P. Funk and N. Xiong, “Extracting knowledge from sensor signals for case-based reasoning with longitudinal time series data,” in *Case-Based Reasoning on images and signals*. Springer, 2008, pp. 247–284.
- [8] A. Aamodt and E. Plaza, “Case-based reasoning: Foundational issues, methodological variations, and system approaches,” *AI communications*, vol. 7, no. 1, pp. 39–59, 1994.
- [9] S. Barua, S. Begum, M. U. Ahmed, and P. Funk, “Classification of ocular artifacts in eeg signals using hierarchical clustering and case-based reasoning,” in *Workshop on Synergies between CBR and Data Mining at 22nd International Conference on Case-Based Reasoning, Cork, Ireland*, 2014.
- [10] N. S. Altman, “An introduction to kernel and nearest-neighbor nonparametric regression,” *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.
- [11] D. E. Robbins, V. P. Gurupur, and J. Tanik, “Information architecture of a clinical decision support system,” in *2011 Proceedings of IEEE Southeastcon*, March 2011, pp. 374–378.
- [12] A. X. Garg, N. K. Adhikari, H. McDonald, M. P. Rosas-Arellano, P. Devereaux, J. Beyene, J. Sam, and R. B. Haynes, “Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: a systematic review,” *Jama*, vol. 293, no. 10, pp. 1223–1238, 2005.
- [13] S. L. Achour, M. Dojat, C. Rieux, P. Bierling, and E. Lepage, “A umls-based knowledge acquisition tool for rule-based clinical decision support system development,” *Journal of the American Medical Informatics Association*, vol. 8, no. 4, p. 351, 2001. [Online]. Available: [+http://dx.doi.org/10.1136/jamia.2001.0080351](http://dx.doi.org/10.1136/jamia.2001.0080351)
- [14] Cerebral pareses – cp. <https://www.1177.se/Vastra-Gotaland/Fakta-och-rad/Sjukdomar/Cerebral-pareses---CP/?ar=True>. Accessed: 2017-05-24.
- [15] R. Palisano, P. Rosenbaum, S. Walter, D. Russell, E. Wood, and B. Galuppi, “Development and reliability of a system to classify gross motor function in children with cerebral palsy,” *Developmental Medicine Child Neurology*, vol. 39, no. 4, pp. 214–223, 1997. [Online]. Available: <http://dx.doi.org/10.1111/j.1469-8749.1997.tb07414.x>
- [16] R. J. Palisano, P. Rosenbaum, D. Bartlett, and M. H. Livingston, “Content validity of the expanded and revised gross motor function classification system,” *Developmental Medicine Child Neurology*, vol. 50, no. 10, pp. 744–750, 2008. [Online]. Available: <http://dx.doi.org/10.1111/j.1469-8749.2008.03089.x>

- [17] G. Häggglund and P. Wagner, “Development of spasticity with age in a total population of children with cerebral palsy,” *BMC Musculoskeletal Disorders*, vol. 9, no. 1, p. 150, 2008.
- [18] G. Ruan, H. Zhang, and B. Plale, “Parallel and quantitative sequential pattern mining for large-scale interval-based temporal data,” in *2014 IEEE International Conference on Big Data (Big Data)*, Oct 2014, pp. 32–39.
- [19] Vad är cpup. <http://cpup.se/vad-ar-cpup/>. CPUP. Accessed: 2017-05-24.
- [20] Vad är cp. <http://cpup.se/vad-ar-cp/>. CPUP. Accessed: 2017-05-24.
- [21] Årsrapport 2016. <http://cpup.se/wp-content/uploads/2017/01/%C3%85rsrapport-enkel-2016.pdf>. CPUP. Accessed: 2017-05-24.
- [22] Bedömnings - och rapporteringsintervall för barn och ungdomar i cpup. <http://cpup.se/wp-content/uploads/2017/01/Bed%C3%B6mningsintervall-barn-i-CPUP-170101.pdf>. CPUP. Accessed: 2017-05-24.
- [23] S. J. Korzeniewski, J. F. Feldman, J. M. Lorenz, J. A. Pinto-Martin, A. H. Whitaker, and N. Paneth, “Persistence of cerebral palsy diagnosis,” *Journal of Child Neurology*, vol. 31, no. 4, pp. 461–467, 2016, PMID: 26271791.
- [24] M. Liedlgruber and A. Uhl, “Computer-aided decision support systems for endoscopy in the gastrointestinal tract: A review,” *IEEE Reviews in Biomedical Engineering*, vol. 4, pp. 73–88, 2011.
- [25] C. S. Lee and M. H. Wang, “A fuzzy expert system for diabetes decision support application,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 1, pp. 139–153, Feb 2011.
- [26] A. Alriksson-Schmidt and G. Häggglund, “Pain in children and adolescents with cerebral palsy: a population-based registry study,” *Acta Paediatrica*, vol. 105, no. 6, pp. 665–670, 2016.
- [27] L. WESTBOM, L. BERGSTRAND, P. WAGNER, and E. NORDMARK, “Survival at 19 years of age in a total population of children and young people with cerebral palsy,” *Developmental Medicine Child Neurology*, vol. 53, no. 9, pp. 808–814, 2011.
- [28] Ibm watson analytics. <http://www-03.ibm.com/software/products/en/watson-analytics>. IBM. Accessed: 2017-05-24.
- [29] D. Henschen, “Ibm watson analytics goes public,” <http://www.informationweek.com/big-data/big-data-analytics/ibm-watson-analytics-goes-public/d/d-id/1317887>, Information-Week, Dec 2014, accessed: 2017-05-24.
- [30] M. Altshuller, “How watson analytics provides a smarter way to work,” <https://www.forbes.com/sites/ibm/2015/01/05/how-watson-analytics-provides-a-smarter-way-to-work/#78dad0b62652>, Forbes, Jan 2015, accessed: 2017-05-24.
- [31] M. Devarakonda, D. Zhang, C.-H. Tsou, and M. Bornea, “Problem-oriented patient record summary: an early report on a watson application,” in *e-Health Networking, Applications and Services (Healthcom), 2014 IEEE 16th International Conference on*. IEEE, 2014, pp. 281–286.