MINISTERUL EDUCAŢIEI NAŢIONALE ŞI CERCETĂRII ŞTIINŢIFICE
UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Technical University of Cluj-Napoca
Faculty of Automation and Computer Science
Computer Science Department

OPTIMIZATION TECHNIQUES FOR ELDERS MALNUTRITION PREVENTION

Master Thesis

Graduate Student
Moldovan Dorin-Vasile

Supervisor
Lect. Dr. Eng. Ionut ANGHEL

Iulie 2016

MINISTERUL EDUCAȚIEI NAȚIONALE ȘI CERCETĂRII ȘTIINȚIFICE
UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Technical University of Cluj-Napoca
Faculty of Automation and Computer Science
Computer Science Department

Dean of Faculty
Prof. dr. ing. Liviu MICLEA

Head of Department
Prof. dr. ing. Rodica POTOLEA

OPTIMIZATION TECHNIQUES FOR ELDERS MALNUTRITION PREVENTION

Master Thesis

1. **Graduate Student**: Moldovan Dorin-Vasile
2. **Supervisor**: Lect. Dr. Eng. Ionut ANGHEL
3. **Thesis contents**: Introduction, Context and Motivation, Thesis Objectives, Bibliographic Research, Architecture of the System, Diets Ontology Model, Reasoning Module, Diets Generation Module, Experimental Results, Conclusions, Bibliography, Acronyms, Relevant Code, Published Papers.
4. **Documentation place**: Technical University of Cluj-Napoca, Computer Science Department, Distributed Systems Research Laboratory (DSRL)
5. **Advisors**: Lect. Dr. Eng. Tudor CIOARA, Prof. Dr. Eng. Ioan SALOMIE
6. **Thesis specifications' date**: ....
7. **Thesis termination' date**: ....

Signature of the Supervisor
Lect. Dr. Eng. Ionut ANGHEL

Signature of the Graduate Student
Moldovan Dorin-Vasile

Iulie 2016

str. Memorandumului nr.28, 400114 Cluj-Napoca, România
tel. +40-264-401200, fax +40-264-592055, secretariat tel. +40+264-202209, fax +40-264-202280
http://www.utcluj.ro/

MINISTERUL EDUCAŢIEI NAŢIONALE ŞI CERCETĂRII ŞTIINŢIFICE
UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Technical University of Cluj-Napoca
Faculty of Automation and Computer Science
Computer Science Department

Author's Declaration Regarding the Autenticity of the master thesis

Subsemnatul *Moldovan Dorin-Vasile*, legitimat cu *CI/BI* seria *KX* numărul *703705*, CNP *1911102125479*, autorul lucrării *OPTIMIZATION TECHNIQUES FOR ELDERS MALNUTRITION PREVENTION* elaborată în vederea susținerii examenului de finalizare a studiilor de masterat la Facultatea de Automatică și Calculatoare, Departamentul Calculatoare, Specializarea *IS* din cadrul Universității Tehnice din Cluj-Napoca, sesiunea *Iulie* a anului univeristar *2015/2016*, declar pe proprie răspundere, că această lucrare este rezultatul propriei mele activități intelectuale, pe baza cercetărilor mele și pe baza informatiilor obținute din surse care au fost citate în textul lucrării și în bibliografie.

Declar că această lucrare nu conține porțiuni plagiate, iar sursele bibliografice au fost folosite cu respectarea legislației române și a convențiilor internaționale privind drepturile de autor.

Declar, de asemenea, că această lucrare nu a mai fost prezentată în fața unei alte comisii de examen de licență sau disertație.

În cazul constatării ulterioare a unor declarații false, voi suporta sancțiunile administrative, respectiv, *anularea examenului de disertație*.

Cluj-Napoca
date

Signature of the Graduate Student
Moldovan Dorin-Vasile

str. Memorandumului nr.28, 400114 Cluj-Napoca, România
tel. +40-264-401200, fax +40-264-592055, secretariat tel. +40+264-202209, fax +40-264-202280
http://www.utcluj.ro/

Abstract

Malnutrition is a serious problem associated with the elder population from all around the world. The aim of the thesis is to propose optimization techniques for the prevention of malnutrition. The proposed optimization techniques vary from the generation of the daily nutrition intake recommendations to algorithms for generating a set of five meals that best match the profile and the preferences of the elders. For the generating of the daily nutrition intake recommendations, four different approaches were analyzed, each of them involving ontologies and databases: KAON2, Jena and D2RQ, Jena and R2RML and OWL API and SQL. For the generating of healthy meals recommendations for an entire day, two bio-inspired algorithms were used: Particle Swarm Optimization (PSO) and Glow-worm Swarm Optimization (GSO). Finally, these algorithms are further optimized using techniques from big data processing, obtaining two modified versions PSOS (PSO using Spark) and GSOS (GSO using Spark). The need for distributing the computational effort on several nodes is justified by the large quantity of input data.

Contents

List of Tables

List of Figures

Listings

List of Algorithms

Chapter 1

Introduction

Malnutrition [4] represents a serious global problem. Estimates show that there are nearly 925 million hungry people in the world. Among the features which influence the hunger and the undernutrition of the people, some of the most important are the security of the food, the household food acquisition, the food intake and the nutritional status. An overview of the problems that exist in relation to malnutrition and the elderly population is also described in [5]. Malnutrition represents the state of being poorly nourished and it can be caused either by the lack of some nutrients or by the excess of other nutrients. Some risk factors for malnutrition are physical disabilities such as arthritis and poor mobility, different disease states such as cancer, lack of knowledge about food, the inability to reach food and lifestyle and social factors. The authors of the paper [6] describe the social and the economical determinants for malnutrition in elderly. They found that there exists a correlation between malnutrition in elderly and some factors such as the inability to shop, the distance from markets and the impossibility to use public transportation. There is a strong relationship between the health and the nutritional status of the elders. In some cases malnutrition is the direct consequence of energy deficiencies. Some consequences of malnutrition are impaired muscle function, anemia, poor wound healing and so on. Elders are often unable to shop and prepare cook meals and this fact causes a vicious cycle between malnutrition and progressive deterioration of senses. Two key messages that can be concluded from this paper are: malnutrition is one of the main causes of mortality in the elderly and the risk of malnutrition can be detected by the screening of the nutritional status.

Malnutrition is viewed as a situation determined by a multitude of factors by the author of paper [7]. Some of the causes of malnutrition are inadequate food intake, choices of food which lead to dietary deficiencies and diseases that cause specific nutrient requirements. The factors that influence the nutritional inadequacy in the elderly population can be classified in four categories: physiologic factors, pathologic factors, sociologic factors and psychologic factors. The author of the paper suggests that the assessment of the nutritional status and the weight loss of the elder should start with the questioning of the elder about any changes in the weight loss during the past three months or past year. Some of the actors that can contribute to the treatment of malnutrition are nurses, dieticians, therapists, and social services staff. The authors of paper [8] describe some concepts related to underweight and undernutrition that correspond to the aging process. The main cause of underweight is represented by the suboptimal intake of calories and other nutrients. Good nutrition is very important as it influences both the good health and the personal productivity of the elders. An insufficient intake of calories leads to the weakening of the immunity system and to exposure to infections. There are different possibilities to measure undernutrition such as the Body Mass Index, the Nutritional Risk Index, the Maastricht Index, and so on. Another aspect described in the paper is represented by the fact that the immune response may be boosted by providing a nutritional supplement with key vitamins.

A discussion about screening and assessment for malnutrition is presented in paper [9]. The screening and the assessment for malnutrition represent an important aspect in order to provide appropriate nutrition intervention. Some of the observations described in the paper are the following: the energy requirements decrease due to the lack of physical activity, malnutrition can affect the well-being of older persons making existing medical problems worse, diets which are very low in fat can produce adverse consequences, and the recommended value for calcium intake is 1500 grams per day. The paper also presents dietary guidelines for healthy Americans: the daily nutrition intake should be composed from a variety of foods, there should be a balance between the physical activity and the foods consumed during the day, the diets should be plenty of grain products, vegetables and fruit, the diets should be low in fat, saturated fat and cholesterol, and the diets should also be moderate in sugars, salt and sodium. Malnutrition, as defined by the author of the paper, represents a state which is produced by one of the following causes: too few macronutrients, too many macronutrients or excessive amounts of inappropriate substances such as alcohol. The screening phase and the intervention phase are critical in order to institute appropriate nutrition intervention.

As stated by the authors of the paper [10], malnutrition represents a serious and a frequent problem in the elderly. The malnutrition in the elderly is defined as inadequate nutritional status, undernourishment, poor appetite, wasting of muscle and loss of weight. The malnutrition is a general term which encompasses several concepts such as: undernutrition, overnutrition, specific nutrient deficiencies and imbalance due to disproportionate intake. Malnutrition is defined as an insufficient dietary intake to meet the requirements for protein needs or for energy needs. Malnutrition in the elderly decreases the quality of the life of the elders and increases the morbidity and the mortality rates. The paper concludes by stating that the malnutrition in the elderly is a multidimensional issue. Another perspective of malnutrition is presented in [11]. Some of the consequences of malnutrition are decreased well-being and deteriorated overall health. There are a lot of important risk factors such as: depression, loss of appetite, stroke, eating dependencies and so on. The development of malnutrition may be prevented and it can also be reversed.

Chapter 2

Context and Motivation

Malnutrition represents a serious problem among the older population from Europe. Some general causes of malnutrition are: suboptimal food intake for some macronutrients, the consumption of foods that are incompatible with the health condition of the elder, and the inability to procure foods in real time. A selection of risk factors [12] for the developing of malnutrition are the depression, the chronic diseases, and the poor dentition. Some of the factors that influence malnutrition are: physical and medical conditions, mental health status, functional status, social domain, economic status and environmental factors. Some common indicators of malnutrition in elderly [13] are involuntary weight loss, abnormal values for body mass index (BMI), vitamin deficiencies and decreased or increased dietary intake.

The aim of the thesis is to describe optimization techniques in order to prevent malnutrition by monitoring and assessing the feeding behavior of the elders, by specifying the corresponding diagnosis when the feeding behavior is inappropriate, by suggesting a combination of foods that the elder should consume in order to reduce the effort of the elder which is necessary for the searching of an appropriate combination of foods that respect his or her nutritional constraints, and lastly to ease the work of the food providers and of the nutritionists.

The main actors involved in the process of optimization are the elders, the nutritionists, the food providers and the informal carers. The relevant information which is necessary for the assessment of the feeding behavior of the older is represented by his anthropometric measurements such as height, weight and body mass index (BMI), his/her gender, age and so on. Based on these measurements, the recommended values of the nutritional values that the elder should consume during the day should be computed. Also, by monitoring the feeding behavior of the elder and by comparing the recommended values with the values obtained after the summation of all the nutrients consumed during the day by the elder, the objective is to display whether the elder respects the feeding constraints associated with him or not. The elder should also have the possibility to command foods from different food providers that respect his dietary constraints and which are in harmony with the diseases associated with him. The elder should be able to consult his daily history of food intake, to record all the foods that he consumes during the day, and to obtain a combination of foods for the entire day. The nutritionist should be able to monitor the feeding behaviour of the elders. He should be able to introduce different diets and to specify different dietary constraints. The nutritionist should also be able to introduce and update a set of parameters which are necessary for the generation of the recommendations for an entire day. The objective of the food providers is to specify the foods that can be consumed by the elders. Also, the products provided by them should be classified according to some diets or diseases, so that the products that are not compatible with the condition of the elders to be omitted from the search space. There should be available dishes for each part of the day: breakfast, brunch, lunch, snack and dinner. Each of the dishes must have a price, a preparation time and ratings for some reliability dimensions. Finally, the fourth category of actors is represented by the informal carers. They should be able to monitor the daily feeding behavior of the elders and to notify a doctor or a nutritionist when the feeding behavior is not respected. They should also help the elders if they can not move. The informal carers represent a link between the elders, the nutritionists and the food providers.

Chapter 3

Thesis Objectives

The main objective of this thesis is to define a model and to propose techniques and algorithms that optimize the problem of generating healthy diets recommendations for the elders by taking into consideration their nutritional constraints and their dietary preferences.

In order to achieve this objective, the next support objectives were defined.

Table 3.1: Secondary Objectives

NR	Secondary Objective	Thesis Approach Description	Achieved in
1	Critical study of the optimization techniques that can be used for the generation of dietary recommendations	Around 100 papers were studied and organized in the following categories: similar approaches, general concepts of the ontologies, tools that can be used to develop ontologies, reasoning using ontologies, tools that can be used to map ontologies and databases, bio-inspired heuristics, PSO algorithm, GSO algorithm and big-data processing using Apache Spark	Chapter 4
2	Description of the architecture of the system used for the evaluation of the proposed optimization techniques	Description of the conceptual architecture of the system and a brief description of the optimization techniques from the following areas: ontologies, bio-inspired heuristics and big data processing	Chapter 5

3	Description of the diets ontology model that is used for the representation of the concepts involved in the optimization techniques	The main concepts of the diets ontology model that are used for the representation of the data involved in the optimization techniques are described by presenting the subontologies that compose the Diets Ontology	Chapter 6
4	Design a reasoning module for the generation of the nutritional recommendations for the elders	Description of the nutritional recommendations associated with the elders and the utilization of the reasoning rules for the representation of the constraints, a critical comparison of the existing mapping tools between the ontologies and the databases and the computation of the score for a custom defined diet	Chapter 7
5	Design a module for the generation of healthy diets for the elders	Description of two bio-inspired heuristics for the generation of healthy diets for the elders based on their nutritional constraints: Particle Swarm Optimization (PSO) and Glow-worm Swarm Optimization (GSO) and description of the same two heuristics using Apache Spark for the optimization of the processing of the data	Chapter 8
6	Conduct a series of experiments using the optimization techniques described above	Compare the different technologies that can be used to map the ontologies and the databases and compare the four approaches used for the generation of the healthy diets for the elders	Chapter 9
7	Describe the conclusions obtained after the comparison of the optimization techniques used for the generation of diets for the elders	Present the conclusions for each optimization technique used in the generation of diets for the elders, describe advantages and disadvantages for using these optimization techniques and present further possible developments	Chapter 10

Chapter 4

Bibliographic research

This section presents a critical overview of the approaches that can be used in order to optimize the malnutrition prevention in elders. The areas that are covered are the following: similar approaches used for the generation of healthy meals recommendations for the elders, ontologies, tools that can be used to develop ontologies, reasoning using ontologies, mapping tools between the ontologies and databases, bio-inspired heuristics, the Particle Swarm Optimization (PSO) algorithm, the Glow-worm Swarm Optimization (GSO) algorithm, and Big Data Processing using Apache Spark.

4.1 Similar Approaches

One approach for generating diets for the elders is presented in [14] and it consists in the use of Genetic Algorithms (GA). This approach described in the paper uses a database of foods and generates a recommendation of five meals for each day of the week by taking into consideration the mentioned restrictions and the preferences for some groups of foods. A technique that uses the Elitist Non-Dominated Sorted Genetic Algorithm in a multi-level way for the generating of dietary menus by taking into consideration several constraints and nutritional requirements in is presented in [15]. One method for the composition of diets for patients suffering from kidney and urinary tract diseases [16] is to use a genetic fuzzy approach which combines fuzzy logic and genetic algorithms. The authors of the paper [17] propose an algorithm for producing a list of food items that meet specific nutritional requirements. Another proposed method for the optimization of the diets is represented by the Quantum Genetic Algorithm (QGA) [18]. The diet optimization method based on QGA gives better results than the traditional algorithms and the Genetic Algorithm (GA). The Bayesian Optimization Algorithm (BOA) was applied by the authors of the paper [19] to the problem of nutrition for breakfast. Compared to other approaches such as Linear Programming and Genetic Algorithm, BOA gives better results.

4.2 Ontologies

One of the ways to model information from the real world is represented by ontologies [20]. Ontologies represent one of the most complex, complete and powerful models for data representation. Ontologies classify not only the classes but also the individuals. They can be represented as directed graphs [21] where the nodes correspond to concepts and the edges correspond to relations between concepts. A more detailed description of ontologies is presented in [22]. Usually, ontologies represent a model of the real world domain such as medicine, geography, physics, e-government and so on. The model is explicitly represented with objects, concepts, entities, and relationships between them. The authors of paper [23] present a survey of the different ways to represent knowledge in context-aware computing environments. The approaches for knowledge representation and inference discussed in the paper are the logic based approaches, the propositional logic, the first order logic, the description logic, the fuzzy logic, the genetic algorithms, the neural networks, the data mining, and the Bayesian networks. Some of the reasons that justify why ontologies are important are described in [24]. An ontology is a representation vocabulary and it is sometimes used to describe a body of knowledge characteristic to some domain. The vocabulary is used to describe the facts associated to a domain while the body of knowledge represents a collection of facts about a domain. In addition to simple facts, an ontology can also represent goals, beliefs or hypothesis. Another statement of the authors of the paper is that ontologies usually appear as a taxonomic tree of conceptualizations. The top of the hierarchy contains very general concepts while the lower levels of the tree describe domain specific concepts. The concept of ontologies has also contributed to the development of the Semantic Web which is an extension of the World Wide Web [25]. The World Wide Web represents the largest database known by humans, but a big disadvantage of it is represented by the fact that even though it is understandable by humans, it is not understandable by machines. Semantic Web represents the new-generation Web and it tries to represent the information such that it can be used by machines. RDF (Resource Description Framework) represents a basic data model that can be used to write statements about Web objects. The three main components of the RDF Model are the resources, the properties and the statements. Some of the advantages of using ontologies are the following: ontologies enable the reusability of the domain knowledge, the development of the ontologies is a cooperative process as it allows many people to describe a given domain, and they allow the separation of the operational knowledge from the domain knowledge. The two different types that are used to describe the types of statements in ontologies are the ABox (Assertion Component) and the TBox (Terminological Component) [26]. The ABox contains membership assertions, attributes assertions, linkages assertions, while the TBox contains definitions of the concepts, classifications and declarations of concept axioms.

The authors of paper [27] also describe what is an ontology. In Computer Science, ontologies represent a special kind of information objects or artifacts used for computations. The backbone of the ontology is represented by a taxonomy which is a generalization hierarchy of concepts. Ontologies describe a body of knowledge that consists of concepts, objects and other entities and the relationships among them. Some of the benefits of adopting Natural Language Generation (NLG) on Semantic Web are reflected by NaturalOWL [28], an open-source tool written in Java. The main objective of the tool is to make the information fully accessible both for humans and for computers. A more detailed description of ontologies is presented in [29]. They represent knowledge representation structures composed of concepts, individuals, properties, relationships among them, and axioms. A concept describes a set of objects that have common properties, and a property is used to describe a relationship between concepts. Individuals represent specific elements of a class, while a relationship is an instance of a property. Axioms, or rules, are used to check the consistency of an ontology and are also used to perform inferences. One of the uses of the ontologies is to annotate scholarly publications and scientific documents. A survey of the ontologies that can be used for this purpose is presented in [30]. The authors of paper [31] describe how to map between ontologies on the Semantic Web. Ontologies make possible the publication of data understandable by machines. Since data comes inevitably from different ontologies, as the nature of Semantic Web is distributed, it is necessary to find mappings between such different sources of data. The solution proposed in the paper is to use machine learning techniques in order to find such mappings. Another approach of mapping between ontologies is presented in [32]. One of the main directions of the Semantic Web is to provide as much automated support as possible for the mapping of different ontologies. The solution proposed by the authors is to create a virtual ontology that consists of the constructs that are mapped from the input ontologies. The ontologies can be used to model the data from various domains. Next, some examples from literature are presented. The authors of paper [33] describe the use of ontologies in the development of countermeasures for military aircraft, the authors of paper [34] describe the use of ontologies in the domain analysis of domain specific languages, and the authors of paper [35] describe a food ontology.

4.3 Tools Used to Develop Ontologies

A tool that can be used for the development of ontologies is Jena API [36]. The core interface around which all the other components are built is represented by the RDF (Resource Description Framework) Graph. Some of the functionalities provided by Jena are: an RDF/XML parser, a query language and RDF/XML output. The RDF graphs can be stored either in memory or in persistent stores. The RDF graph is a collection of Triples, and each Triple comprises three parts or Nodes: the subject, the predicate and the object fields. One of the disadvantages of Jena is represented by the fact that the query processing performance is not acceptable. Jena stores the information in a single table and this is the main reason Jena storage consumes too much time [37]. A comparison between Jena API and Protege is presented in [38]. Jena is a Java framework used for the building of Semantic Web applications and it provides a programming environment for RDF, RDFS, OWL (Web Ontology Language) and SPARQL. Jena classes can be used to parse an RDF file and to store it in memory. On the other hand, Protege is specially designed for loading and saving OWL and RDF ontologies, for the declaration of SWRL (Semantic Web Rule Language) rules and for the editing of OWL individuals. Jena is a general RDF/RDFS framework while Protege API is dedicated to OWL manipulation. As its central interface, Jena offers an abstraction of the RDF graph [39]. There are two types of properties supported by the Jena framework [40]: object properties describe relationships between two instances of ontology classes and datatype properties describe the data type of a class instance, including both a domain and a range. Other features of Jena API are described in [41]. The core of the Jena API consists of a model and common classes. The input data is processed by an XML parser and the output data is written by an XML writer. The information can be stored in a memory store, an SQL store, a prolog store and so on. Data can be processed using a query engine. One of the goals of ontologies is to make the content on the web accessible and machine understandable [42]. The three layers of Semantic Web are Resource Description Framework (RDF), RDF Schema and Web Ontology Language (OWL). Jena has classes to represent graphs, resources, properties and literals and these classes are named Resource, Property and Literal. Another characteristic of Jena is represented by the fact that it can infer relationships from a previously defined model using a Reasoner.

Another API that can be used for the manipulation of the ontologies is the OWL API [43]. The functions provided by the API are the following: serializing, modelling, parsing, manipulation, and inference. Other features of the OWL Application Programming Interface are [44]: it supports the manipulation of ontological structures, it supports the use of reasoning engines, and it supports parsing and rendering in the syntaxes defined in the W3C specification. An ontology represents a collection of annotations and axioms. Some basic design principles that endured are the following: there is a clear separation between inference and assertion, there is a clear separation between components that provide different functionalities, and it provides independence from concrete serializations. The OWL API also provides support for reading and writing ontologies in several syntaxes. An introduction of the OWL API is presented in [45]. OWL allows the description of a domain in terms of individuals, classes and properties. An ontology is defined as a collection of axioms that assert information about classes, individuals or properties. The OWL API provides classes for the representation of OWL ontologies, and classes for creation, manipulation, parsing, rendering, and reasoning on those structures. There is a clear distinction between object properties and data type properties. Object properties are used to relate two individuals and data properties are used to relate an individual with a data value. Also, properties can be associated with ranges and domains. Another feature of the ontologies is represented by the inference. The semantics of OWL allow the possibility to perform reasoning or inference over an ontology.

Ontologies can be queried using SPARQL [46] which represents a graph-matching query language. The authors of paper [47] describe some issues that can appear in the implementation of SPARQL Update which represents an extension of the first version of

SPARQL. Using SPARQL Update, an RDF graph can be updated with new triples. The three operations supported by SPARQL Update are [48]: deletion of data, insertion of data and deletion/insertion of data.

4.4 Reasoning Using Ontologies

One major functionality of the ontologies is represented by the reasoning [49]. The authors of the paper [50] describe a comparison of some reasoners. The standard description logic reasoning tasks are the checking of the consistency of an ontology and the drawing of new conclusions about the knowledge base. The TBox reasoning tasks are the satisfiability and the subsumption while the ABox reasoning tasks are the consistency, the instance checking, the retrieval problem, the property filters, and the conjunctive queries. The features of the reasoners such as the checking of the inconsistencies and the extraction of new knowledge from knowledge that already exists are also described in [51]. Some limitations of SWRL (Semantic Web Rule Language) are presented in [52]: disjunctions and negations are excluded, there are no quantifiers, there are no user defined functions and it uses only assertions. The reasoning steps are the following [53]: the finding of a reasoner, the applying of the reasoner to data, the creating of an inference model, and the accessing of the inferences. The authors of the paper [54] also describe some features of the reasoners. The context-aware computing can be used to solve any inconsistencies that exist in the representation of the context knowledge. The run time performance of the reasoning depends on three factors: the size of information, the speed of the CPU and the complexity of the reasoning rules. A short description of the KAON2 reasoner is provided in [1]. The algorithm provides good performance for ontologies that have complex ABoxes, but simple TBoxes. On the other hand, if the ontologies have complex TBoxes then the reasoner does not provide improvements. There is a linear relationship between the time needed to perform an inference and the energy necessary to complete an inference [55].

4.5 Mapping Tools between Ontologies and Databases

One of the main challenges when using both ontologies and databases for the storage of the information is represented by the mapping between the two ways of data representation. A tool that can be used to combine databases and ontologies is D2RQ [56]. In the case of using D2RQ [57], the ontology classes are mapped to database tables, the ontology individuals are mapped to the rows from the database tables, the data type properties are mapped to the columns, and the object properties are mapped to the foreign keys. The authors of the paper [58] describe several problems related to the transformation between relational databases and ontologies: sometimes the transformed data does not contain the same information as the original data, sometimes there is a loss of semantics, and sometimes the transformation does not have provable correctness. The steps for mapping relational databases to RDF are described in [59]. The first step is represented by the extraction of the semantic information, the second step is represented by the schema mapping, and the third step is represented by the data mapping. The problem of generating and storing of an RDF graph is also discussed in [3]. There are two directions that are presented: the incremental transformation and the incremental storage. The authors of the paper [60] describe several benefits for the mapping between the ontologies and the databases: the mapping between the two approaches for the representation of data can be used to semantically annotate web pages, it is possible to integrate heterogeneous databases, the data can be accessed using ontologies and it is possible to integrate database content with information that comes from other data sources. An approach for mapping between ontologies and databases is also discussed in [61]. In this approach the ontology and the database are deployed separately. A three phases databases mapping methodology based on techniques for ontology mapping is described in [62]: the first phase generates the ontological representation from database schema, the second phase uses matcher algorithms to identify relations between tables and classes, and the third phase verifies the correctness of the mappings. The authors of the paper [63] propose a two phases approach to map a relational database schema to an ontology: the first phase is represented by the searching of correspondences between the relational database schema and the ontology and the second phase is represented by the constructing of more complex compositions. Another idea is presented in [64], where the authors suggest that the mapping process between ontologies and relational databases should be an automatic process. A survey of RDB to RDF translation approaches and tools is presented in [65]. The authors compare seventeen RDB-to-RDF approaches and the main conclusions inferred from this comparison are: some of them support the R2RML W3C recommendation while others have their own mapping language, regardless of the method used for the mapping, the result is always a read-only web of data, and the creating, updating and deleting of data should be made possible in a secure way. An approach based on HTML forms is presented in [66]. The reasons for analyzing the HTML forms are: they are designed to be user-friendly, they do not necessarily reflect the structure of the database and they are the most popular interface to communicate with a relational database. RDB2Onto [67] is another tool that can be used to convert data from a relational database to an RDF/OWL ontology. Its performance is better than the performance of the RDQuery tool but worse than that of the D2RQ tool.

4.6 Bio-Inspired Heuristics

A hot topic in the development of algorithms inspired from nature is represented by the swarm intelligence algorithms and the bio-inspired algorithms [68]. There can be several sources of inspiration such as the swarm intelligence, the biological systems, the physical systems and the chemical systems. Many applications deal with NP-hard problems because the optimization problems from the real world are very difficult to solve. There are no efficient ways to solve NP-hard problems and thus many problems are solved by trial and error and by using various optimization techniques. In the current literature there are approximately 40 different algorithms. Computing has been inspired by nature from the beginning [69]. The biologically inspired computing links many disciplines such as artificial intelligence, agent-based systems and artificial life. Bio-inspired algorithms often work well even though the desired task to solve is poorly defined and they adapt to the changes in the environment of the task. Sometimes it is almost impossible to map a bio-inspired algorithm. An example is given in [70] where a comparison between the communication among fireflies and the communication among sensors is made. In real life, the fireflies communicate with the speed of light while the sensor systems communicate slower due to latencies. Bio-inspired algorithms can be applied to problems from various domains. Another example is presented in [71] where the particle swarm optimization (PSO) and the bacterial foraging algorithm (BFA) are used to localize nodes in wireless sensor networks. The parameters taken into consideration for the comparison of the two approaches are the following: the number of the localized nodes, the accuracy of the localization and the computation time. The authors of the paper [72] use four bio-inspired algorithms in order to perform web service composition: Ant Colony Optimization (ACO), Genetic Algorithm (GA), Evolutionary Algorithm (EA), and Particle Swarm Optimization (PSO). Some of the similarities between the biological systems and the service oriented systems are: the scalability, the adaptability, the robustness, and the reliability. There are two branches for solving classical problems [73]: the exact methods and the heuristics. The exact methods include logical methods and mathematical programming while the heuristics include methods that find solutions when the exact methods fail. Bio-inspired algorithms are a subset of the second category. The design of a bio-inspired algorithm is composed from three steps: the choosing of the representation of the problem, the evaluating of the quality of the solution using a fitness function and the defining of the operators that are necessary to produce a new solution. There are three directions in the area of bio-inspired engineering: swarm intelligence, evolutionary algorithms and the artificial immune system [74]. The authors of the paper [75] use two algorithms for the thresholding of the terrain images: particle swarm optimization (PSO) and bacterial foraging algorithm (BFA). Both of them are faster than the exhaustive approach. The BFA-based localization has a better accuracy compared to the PSO-based localization which is faster. Some of the advantages of the swarm intelligence are also described in [76]: scalability, fault tolerance, adaptation, speed, modularity, autonomy, and parallelism. One of the most used mechanisms for the handling of the constraints is the set of feasibility rules [77]: between two feasible solutions, the one with the highest fitness value is better, between a solution that is feasible and a solution that is not feasible, the feasible solution is better, and between two unfeasible solutions, the one that has the lowest sum of constraint violation is better.

4.7 Particle Swarm Optimization

Particle swarm optimization (PSO) algorithm is a bio-inspired heuristic that optimizes a problem by improving a candidate solution. The problem is represented as a set of candidate solutions which are called particles and which have a position and a velocity. These particles are moved in a search space and each particle's new position and velocity are influenced by the best position of the particle encountered so far and by the best position encountered so far by any particle from the swarm. There are many variations of the particle swarm algorithm. Next, some of these variations are presented. The authors of the paper [78] propose a discrete binary version of the improved particle swarm optimization algorithm. A discrete optimization problem can thus be solved using a binary mixture of the particle swarm algorithm. Also, the authors propose the integration of the simulated annealing in the evolutionary process in order to avoid the cases when the particle is trapped in a local optimal solution and the particle evolves randomly from that point. One of the drawbacks of PSO is represented by the premature convergence of the particles. A solution for this problem is presented in [79], where the selection strategy is based on antibody density and the initiation process is based on equal probability chaos. By applying these changes to the original PSO algorithm, a new algorithm is obtained. This new algorithm is called chaos immune particle swarm optimization algorithm (CIPSO). Another optimization of the PSO algorithm is presented in [80]. The iterative improvement strategy (IIS) can be used to guide the position and the velocity of a particle more finely. With these adjustment, the intensification ability of the PSO algorithm is increased. The robustness and the speed of convergence of the PSO can also be improved by another version of the algorithm which is called phase-angle encoded and quantum-behaved particle swarm optimization [81]. The authors of the paper [82] use the "catfish effect" in order to define a new variant of the PSO which is called the catfish effect particle swarm optimization (CE-PSO) algorithm. This new algorithm has a more powerful global search capability and a higher speed of convergence. The problem of premature convergence of the swarm is also discussed in [83]. During the later phases of the algorithm many particles get closer to the best particles, the activity of the particles is low and the swarm is prone to premature convergence. The proposed solution is to adjust the density of the swarm by establishing the density of the swarm using information entropy and by determining the trends of the particle density. A novel approach for solving a discrete optimization problem is proposed in [84]. In the case of using the multi-state particle swarm optimization (MSPSO) algorithm to solve the optimization problem, the next state can have more than one or two possible states. However, if the cost between the current state and the new generated state is greater than the velocity value obtained after the update of the velocity then the new generated state can no longer be considered a candidate state for the particle. Another version of PSO is presented in [85]. This version is called two subpopulation particle swarm optimization (TSPSO), and it uses two subpopulations to cover the search space. The weights for the cognitive component and for the social component are different for each subpopulation. The first subpopulation has a higher value for the social component and a lower value for the cognitive component, while the second subpopulation has a lower value for the social component and a higher value for the cognitive component. The overall performance of the algorithm is better than that of the classical version of the PSO. A version of the algorithm that uses the global-local optimal information ratio is presented in [86]. The global-local optimal information ratio PSO (GLIR-PSO) is better than the classical version of the PSO in terms of the speed of the convergence of the particles, the quality of the solution and the robustness. Finally, another version of the PSO is presented in [87]. The vector-evaluated particle swarm optimization (VEPSO) uses a multi-swarm model. Each swarm from the multi-swarm model optimizes the problem and the passing of the information between the swarms is realized using a knowledge transfer strategy (KTS).

4.8 Glow-worm Swarm Optimization

The Glow-worm Swarm Optimization (GSO) algorithm [88] is an algorithm inspired by the biological behavior of the glowworms. The glowworms that are brighter attract other glowworms that are less brighter than them. The basic idea of the algorithm is presented next. At the beginning of the algorithm each glowworm is assigned with a random position in the search space and the corresponding luciferin value associated with them depends on the position. The attraction is greater if the glowworm is brighter. At the end of the algorithm, the glowworm warm gathers around the glowworms that are brighter. An algorithm that uses GSO and big data analysis techniques is proposed in [89]. The MapReduce Clustering Glowworm Swarm Optimization (MRCGSO) algorithm handles big data. The MapReduce methodology is used because it provides load balancing and fault tolerance. One of the drawbacks of the classical version of the GSO algorithm is represented by the fact that it has slow convergence rate. The alternative is to use a modified version, the Quantum-behaved Glowworm Swarm Optimization (QGSO) [90]. The QGSO evolves faster than the existing competitors and it provides a better balance between exploitation and exploration. The GSO algorithm can also be used to solve multi-objective optimization problems (MOP-GSO) [91]. Usually, a multi-objective optimization problem (MOP) consists of m objective functions, n decision variables and k constraints. Another improvement of the GSO is discussed in [92] where the author proposes a self-adapting glowworm swarm optimization in which the radial sensor range is adapted with iteration time. The authors of the paper [93] propose a new movement rule. The major drawback of GSO is the fact that it can easily fall into local optimum and in the latter period of the algorithm the values of the accuracy and of the speed of convergence are low. The proposed solution is to use Tent map of chaos to deploy the glowworm and to use another movement rule. An alternative is to use a glowworm swarm optimization algorithm with improved movement pattern (IMGSO) [94]. The convergence is improved by adding global information and adaptive step in the movement phase. The Glowworm Swarm Optimization algorithm has applications in many areas such as: the job shop planning problem [95], the range expansion of mobile wireless system by cooperative transmission [96] and multi-spectral satellite image classification [96].

4.9 Big Data Processing using Apache Spark

Apache Spark [97] is an open-source platform used for large-scale data processing. One of the features of Apache Spark is represented by MLlib, which is an open-source distributed machine learning library. MLlib contains many statistical, linear algebra and optimization primitives. Some of the implementations of the standard learning algorithms which are included in this library are the regression, the classification, the clustering and the collaborative filtering. The core features of MLlib are the following: it supports methods and utilities such as various linear models, or naive Bayes, it provides algorithmic optimizations, it supports a pipeline API, and it can be integrated in Spark. The performance of Spark depends on Java Virtual Machine (JVM) parameters and on Operating System (OS) parameters [98] because even though Spark is an in-memory-oriented computing framework, it runs on top of JVMs. The core of Spark is represented by the resilient distributed datasets (RDDs), which contain a collection of Java objects which is partitioned over a set of nodes. An RDD can be transformed into another RDD by applying a set of operations such as: map, count, filter, etc. The authors of the paper [99] use Apache Spark in order to improve the Smith-Waterman (SW) algorithm. By using the MapReduce framework of the Apache Spark and the Hadoop Distributed File System (HDFS), the obtained algorithm achieves load-balancing for the parallel adaptation. Some of the benefits of using Spark on this algorithm are the following: the algorithm is easy to deploy, the cost of maintenance is low, the algorithm is reliable and the proposed solution provides load-balancing. The two main elements of a Spark cluster are the Master and the Worker [100]. Usually, a cluster is composed from several machines which are also called nodes, and a master node is used to distribute jobs to the slave nodes. Apache Spark can also be used in the analysis of Mobile Big Data [101]. The benefits introduced by Spark are the partitioning of the volume by dividing the main learning task into sub-tasks and the enabling of fast processing of the data using a high-throughput. The description of the execution engine of Spark is better described in [102]. When an action is run on an RDD, the transformation dependencies generate a directed acyclic graph. There are two types of operations that can be performed by the RDDs [103]: actions and transformations. The actions execute computations over a dataset and return a value, while the transformations create new datasets from existing datasets. Three of the most used operations are the mapping, the filtering and the joining [104]. Other benefits of Spark and Hadoop are the possibility to explore massive datasets and the possibility to extract valuable information from these datasets with high performance [105]. Apache Spark not only supports map/reduce operations but it also provides fault tolerance. More details about Spark can be found in [106]. Some drawbacks of MapReduce are the iterative jobs and the interactive analysis. Spark provides an alternative for these two types of operations. The Spark's programming model is centered around three main components: the resilient distributed datasets (RDDs), the parallel operations, such as reduce, collect and foreach, and the shared variables which are of two types, accumulators and broadcast variables.

The key concepts of Apache Spark are also presented in [107]: the resilient distributed datasets (RDDs) and the directed acyclic graph (DAG). The RDDs are the abstraction of Spark for the manipulation of distributed in-memory data, while the DAG execution engine represents the transformation dependencies associated with an RDD. Some benefits of the RDDs are presented in [108]. In the case of failures only the RDD's lost partitions must be recomputed and their immutability allows the system to mitigate the slow nodes using an approach similar to that used by MapReduce [109]. The three main benefits of Spark are also presented in [110]: it is very easy to use, it provides complex algorithms and it is a general engine which allows the combination of computations from different sources. Another application of Apache Spark is presented in [111]. Apache Spark is used in this case in order to construct a parallel framework for the analysis of the DNA. The main benefit of using Spark is represented by the use of the in-memory caching. The dynamic load balancing is an example of such optimization introduced by Spark. Between the execution of the distributed operations, the program data is maintained in memory [112]. The tasks are scheduled using a delay scheduling algorithm which is described in [113]. A description of how Spark works is also presented in [114]: the first step is represented by the creation of a dependency graph from a programming model and the second step is represented by the using of this generated graph to schedule the work and to transport the data and the code to the worker nodes. One of the challenges for Big Data processing is discussed in [115]. There are a lot of different sources of information which range from sensors to messages from social media and this data is represented and interpreted in many ways. Thus, one of the big challenges for Big Data processing is represented by the heterogeneity of the data sources. Finally, a comparison between MapReduce and Spark is presented in [116]. Both of them are open source computing frameworks and can be used for the analytics of data. The experiments conducted by the authors of the paper show that Spark has a better time performance than MapReduce.

Chapter 5

Architecture of the System

The following subsections present the conceptual architecture of the prototype that was used for the simulation of the optimization techniques that are used for the generation of meals recommendations for the elders, a brief description of the ontologies and of the benefits added by the ontologies, a short description of the bio-inspired meta-heuristics and a motivation of why big data processing techniques can be used in this context.

5.1 Conceptual Architecture

The conceptual architecture of the system used for the simulation is presented in Figure 5.1. The main three actors depicted in the figure are the food providers, the elders and the nutritionists. The food providers can insert foods in the database, the elders can introduce their personal information such as height, weight, gender, food preferences, price preferences and so on, and the nutritionists can define nutritional rules that are used further in the process of determining which combination of foods fits best with the nutritional constraints of the elders. Data is represented in two ways: database and ontologies. The database contains information about the foods and about the elders while the ontology describes the model used in the simulation and contains the nutritional rules defined by the nutritionist. In order to generate a combination of foods that matches the nutritional restrictions of the elders and the recommendations specified by the nutritionists, two modules are designed. The reasoning module is used to obtain the nutritional recommendations for the elders based on the reasoning rules and on the personal information of the elders, while the diets generation module is used to search a combination of foods for an elder based on the dietary recommendations obtained from the reasoning module and the information retrieved from the database which contains data about the foods, the preferences of the elders and in some cases the values of some parameters used in the optimization techniques. For the reasoning module, there are four possible approaches which are presented in this thesis: KAON2, Jena and D2RQ, Jena and R2RML and OWL API and SQL. For the diets generation module there are also four methods presented: Particle Swarm Op-

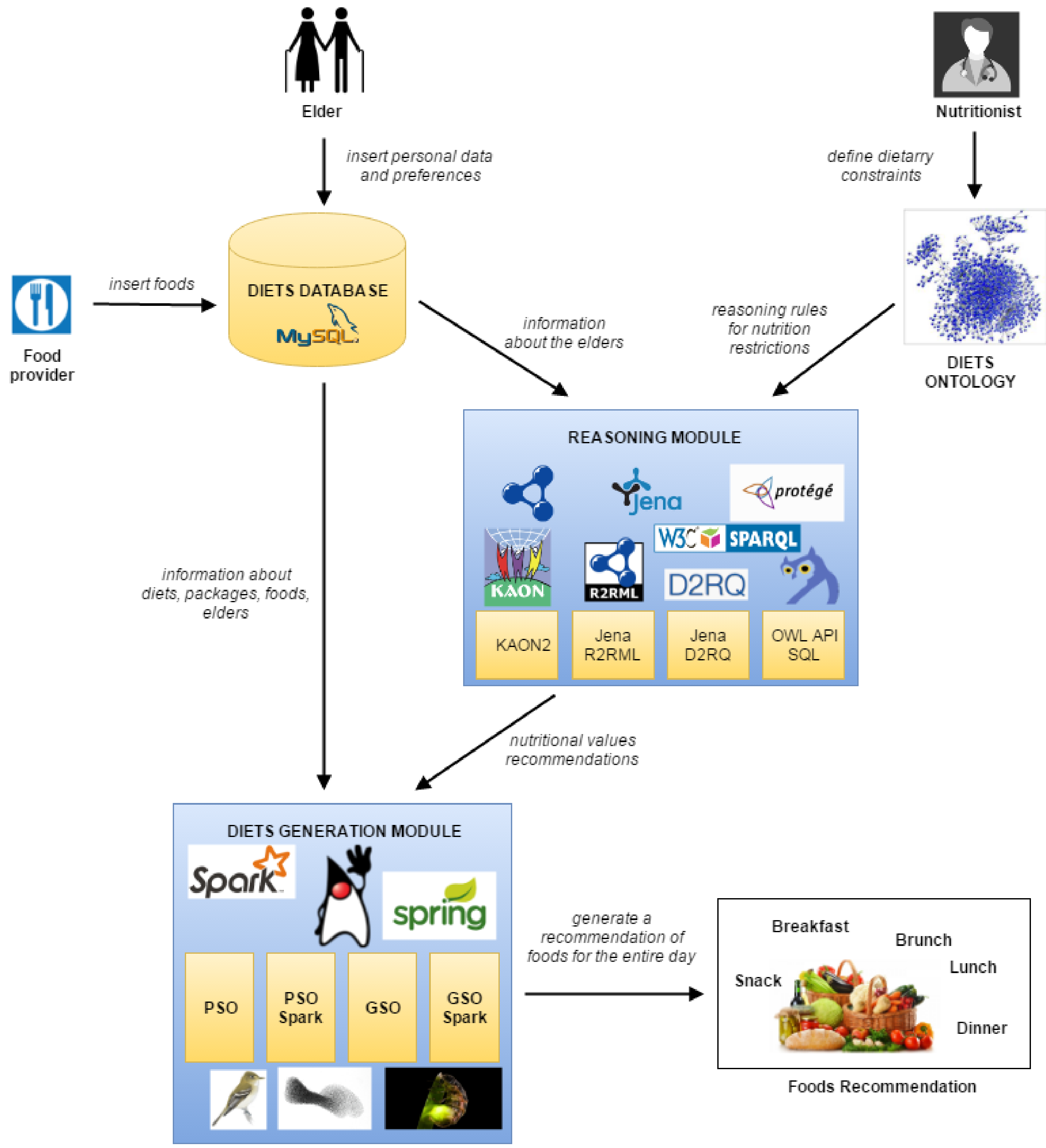

Figure 5.1: Conceptual architecture of the system

timization (PSO), Particle Swarm Optimization using Spark (PSOS), Glow-worm Swarm Optimization (GSO) and Glow-worm Swarm Optimization using Spark (GSOS).

5.2 Ontologies

Ontologies represent one of the most complex ways that can be used to model the data. Considering the fact that data comes from different domains such as nutrition, food providers, anthropometric measurements, diets, diseases, medication, and so on, it is really important to represent the data so that it can be modified easily by the users of the application. By using ontologies, the information can be represented as a graph where the nodes represent individuals or classes, and the edges represent data type properties or object properties. The information can be represented as a tree, where each node from the tree corresponds to a class. The instances of the classes correspond to ontology individuals and as a comparison, the ontology individuals are similar with the instances from the Java programming language. Data type properties can be used to describe the ontology individuals and can have primitive values, while object properties link ontology individuals. The representation of data in this format is not the only advantage of using ontologies. The ontologies can also be used to check whether the representation of the model is consistent or not.

Another aspect related to ontologies is represented by how the data about the individuals is stored. The individuals can be stored either in ontology files or in a database. If the information about the individuals is very large, then the preferred solution is to store the data in a database. In this way, the data can be manipulated in different ways, and thus there is a clear separation between the part of the ontology that is used to describe the model and the part of the ontology that is used to represent the instances. There are different technologies which provide a mapping between an ontology and a relational database, and one direction of the project is to study these technologies, and to see how they apply to the proposed use case. Ontologies can also be used to infer new information from information that already exists in their representation. The information can be processed by ontology reasoners based on predefined reasoning rules. The reasoning rules are expressions of the form "if ... then ..." which can be used to generate additional information from information which already exists in the ontology. An advantage of the usage of the reasoning rules is represented by the fact that in this case the ontologies can use a smaller number of concepts to represent the data.

5.3 Bio-Inspired Heuristics

The problem of generating a recommendation of foods for the entire day represents a very complex problem as the search space is very large. The proposed solution is to use bio inspired meta heuristics that imitate the way in which the nature behaves. The main advantage of the usage of the meta heuristics is represented by the fact that the time necessary to perform the computations is reduced significantly. On the other hand, one disadvantage is represented by the fact that the generated solution is not the best possible solution even though it is an optimal solution. Also, there must be a trade off between the time which is necessary to perform the computations and the quality of the solution. There are different metrics that can be used to characterize the solutions such as the fitness function. The fitness function describes the quality of a generated solution in comparison with the best possible solution. Depending on whether the score obtained after the application of the fitness function on a solution is high or low, it is easy to check whether the generated solution is a good solution or a bad solution.

5.4 Big Data Processing

Another direction of the project is to perform big data processing. If the quantity of information is very large then a single machine is not enough to process all this information and to perform all the computations. In this case the effort may be distributed among a number of different machines. A classic approach to process this information is to use a series of map-reduce operations. The map phase represents the phase in which the operations are mapped on the input data, while the reduce phase is the phase in which the resulting data is collected. In the case of using a single machine, the effort can be distributed among a set of different workers, and the final results can be collected at the end of the execution of the operations.

Chapter 6

Diets Ontology Model

The information is represented in three forms in the simulation prototype: the relational database form, the Object Oriented Programming (OOP) form and the ontological form. The ontological form is the most complex representation from the three ways of data representation. In order to describe the model only once, the ontological representation of the domain was chosen.

The *Diets Ontology* presented in Figure 6.1 is organized in seven subontologies: the *Food Ontology*, the *Food Provider Ontology*, the *Meals Reliability Ontology*, the *Nutrition Behavior Assessment Ontology*, the *Nutrition Behavior Monitoring Ontology*, the *Nutrition Diagnostic Ontology* and the *Nutrition Intervention Ontology*. The subontologies that compose the *Diets Ontology* are adapted after the ones used in [117].

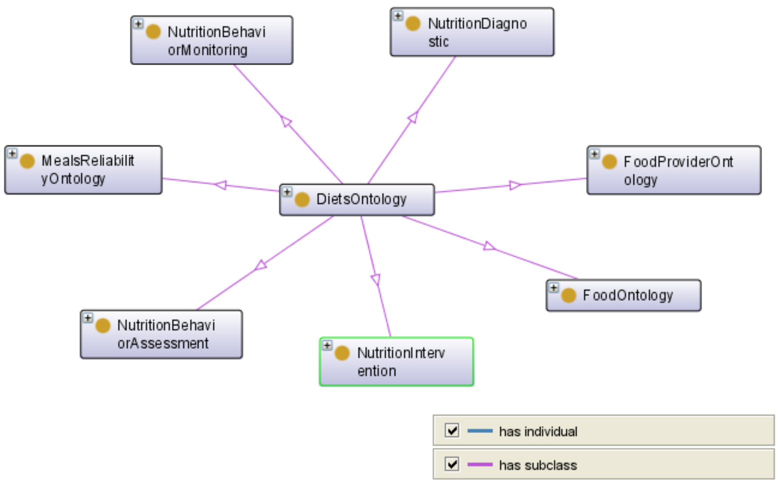

Figure 6.1: Diets Ontology

6.1 Food Ontology

The *Food Ontology* is presented in Figure 6.2. This subontology provides a classification of the foods according to [118].

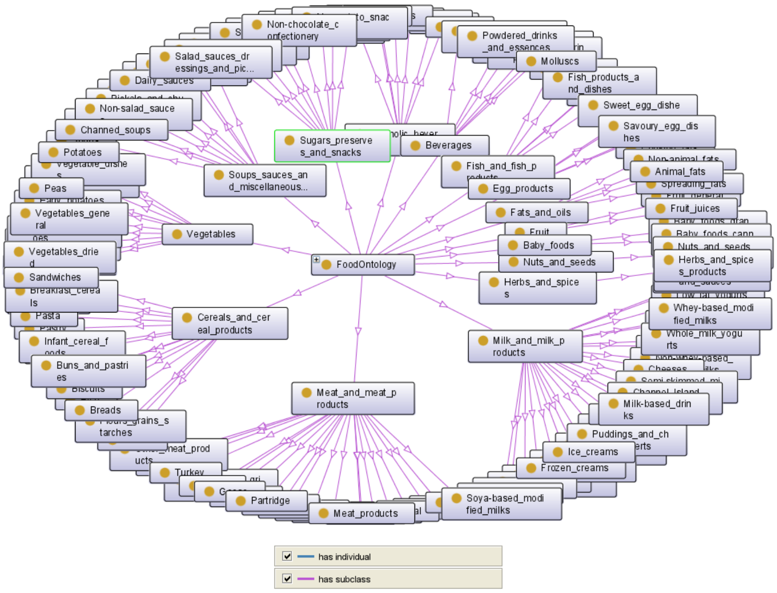

Figure 6.2: Food Ontology

6.2 Food Provider Ontology

Figure 6.3 describes only a subpart of the *Food Provider Ontology*. The main concepts described by this subontology are the meal types (e.g. *Breakfast*, *Brunch*, *Lunch*, *Snack*, *Dinner*), the types of the cycle menus (e.g. *weekly* or *daily*) and the classification of the diets (e.g. *Halal*, *Vegetarian*, *TextureModified*).

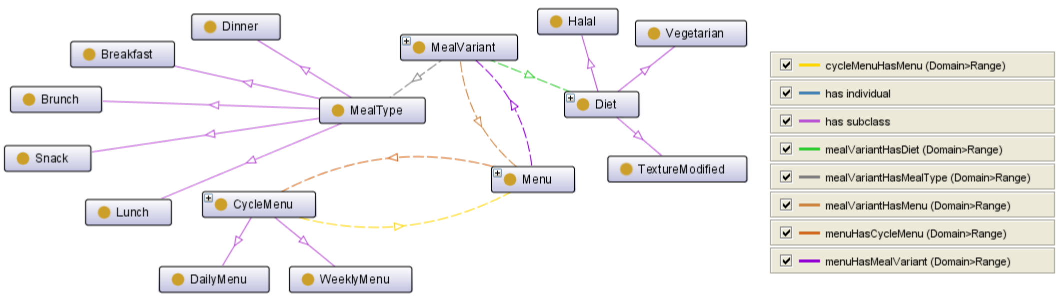

Figure 6.3: Food Provider Ontology

6.3 Meals Reliability Ontology

A part of the *Meals Reliability Ontology* is presented in Figure 6.4. There are three main aspects that are taken into consideration: the delivery aspect, the food quality aspect and the nutrition aspect.

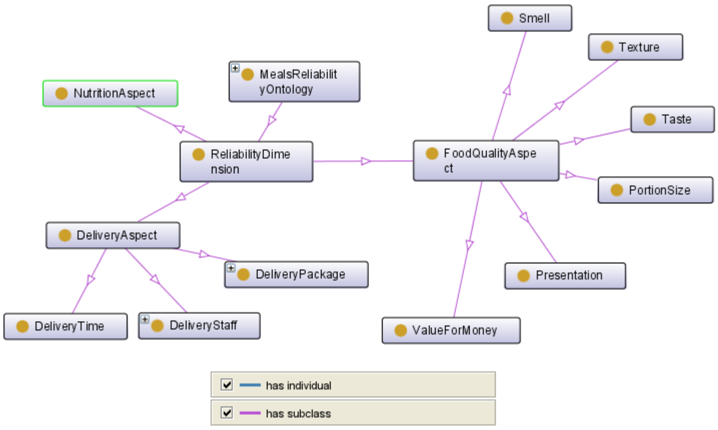

Figure 6.4: Meals Reliability Ontology

6.4 Nutrition Behavior Assessment

The *Nutrition Behavior Assessment Ontology* described in Figure 6.5 contains the main concepts which are used in the daily nutrition evaluation of the nutrition behavior of the elders. There are two types of foods: basic foods and combined foods and there are different food quantity types such as teaspoon, tablespoon, cup, can or bowl.

6.5 Nutrition Behavior Monitoring

The scope of the *Nutrition Behavior Monitoring Ontology* presented in 6.6 is to organize the values that are measured: the anthropometric measurements of the elders, the health profile (e.g. diseases, allergies), the value of the Physical Activity Factor (PAF) and the score to a particular diet such as a Mediterranean Diet.

6.6 Nutrition Diagnostic

The *Nutrition Diagnostic Ontology* presented in Figure 6.7 describes the diagnostics which are relevant for the prevention of the malnutrition.

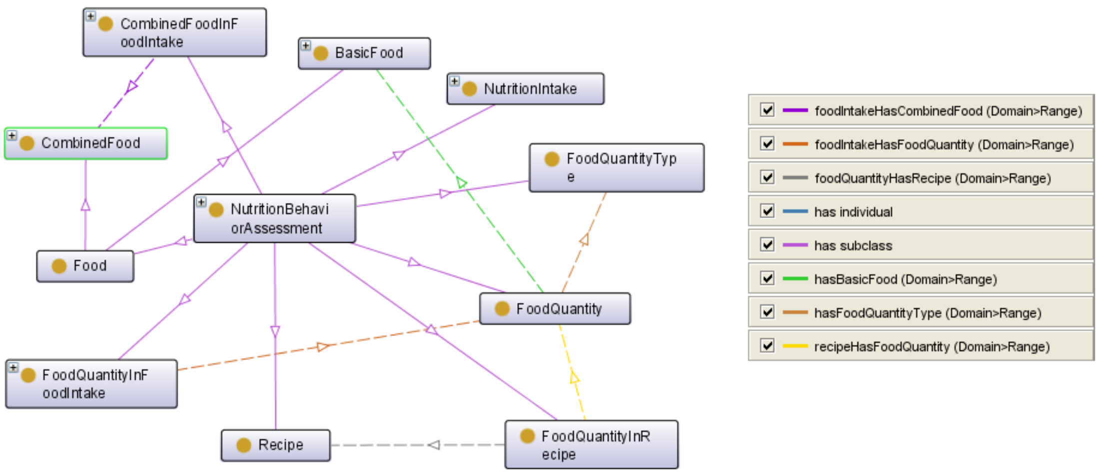

Figure 6.5: Nutrition Behavior Assessment Ontology

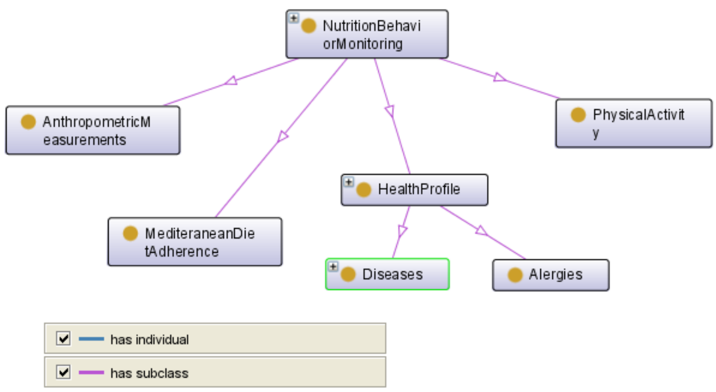

Figure 6.6: Nutrition Behavior Monitoring Ontology

6.7 Nutrition Intervention

The *Nutrition Intervention Ontology* described in Figure 6.8 presents several approaches that can be used in order to prevent malnutrition: nutrition prescription, nutrition education and food ordering.

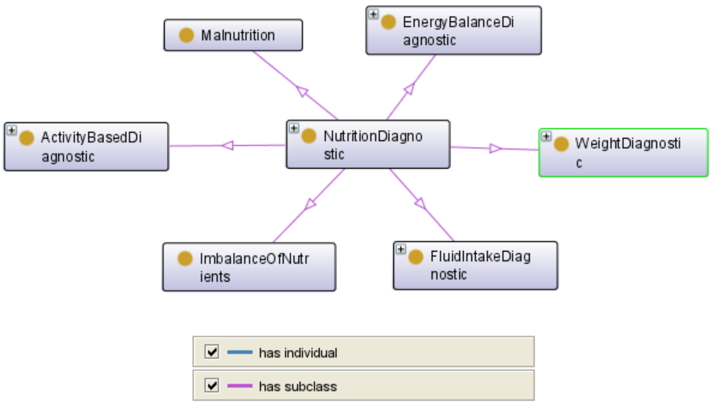

Figure 6.7: Nutrition Diagnostic Ontology

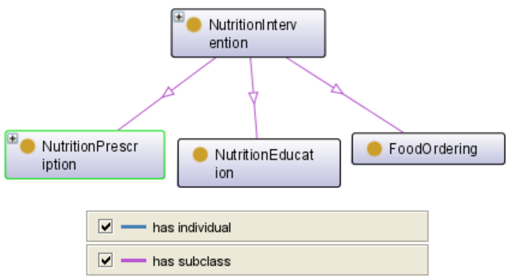

Figure 6.8: Nutrition Intervention Ontology

Chapter 7

Reasoning module

This section presents a description of the reasoning module used in the process of obtaining a recommendation of healthy meals for the elders based on their nutritional constraints. The first subsection presents the mathematical description of the nutritional constraints that are used in order to obtain the nutritional recommendations for the elders, the second subsection presents a critical comparison of the existing mapping tools between ontologies and databases and the third subsection describes the approach for computing the score for a Mediterranean diet.

7.1 Nutritional recommendations as reasoning rules

This subsection describes the mathematical representation of the nutritional recommendations that are used in the process of determining which combination of foods fits best with the profile and the preferences of the elder. These nutritional recommendations are used further in the form of reasoning rules and are processed by a reasoner. The result of their processing represents the numerical values of the nutritional values that the elder should consume during a single day.

7.1.1 Energy (kcal)

The value of the energy is measured in kcal. This value depends on the height of the elder, the weight of the elder and the Physical Activity Factor (PAF) [119]. The PAF value describes a person's daily physical activity as a number. There are different recommended values for men and for women. The optimal energy value that an elder should consume during a day, both for the men and for the women, is computed as the average between the lower energy value that the elder should consume during the day and the upper energy value that the elder should consume during the day. The optimal energy value that an elder should consume during the day is used further in the process of calculating the daily optimal values of other nutrients.

$$Estimated\ Energy\ Value\ (men) = (((weight \times 0.0478) + (height \times 0.0257)) - 1.07) \times PAF \times 239 \quad (7.1)$$

$$Estimated\ Energy\ Value\ (women) = (((weight \times 0.0356) + (height \times 0.0176)) + 0.0448) \times PAF \times 239 \quad (7.2)$$

$$Estimated\ Energy\ Value\ Lower\ Limit = Estimated\ Energy\ Value \quad (7.3)$$

$$Estimated\ Energy\ Value\ Upper\ Limit = 110\% \times Estimated\ Energy\ Value \quad (7.4)$$

$$Energy = \frac{Estimated\ Energy\ Value\ Lower\ Limit + Estimated\ Energy\ Value\ Upper\ Limit}{2} \quad (7.5)$$

7.1.2 Carbohydrates (g)

The value of carbohydrates that an elder should consume during the day varies between 45% and 65% from the optimal energy value divided by 400. The optimal value of carbohydrates is expressed in grams and is computed as the average of these two limits.

$$Estimated\ Carbohydrates\ Value\ Lower\ Limit = \frac{45 \times Energy}{400} \quad (7.6)$$

$$Estimated\ Carbohydrates\ Value\ Upper\ Limit = \frac{65 \times Energy}{400} \quad (7.7)$$

$$Carbohydrates = \frac{Estimated\ Carb\ Value\ Lower\ Limit + Estimated\ Carb\ Value\ Upper\ Limit}{2} \quad (7.8)$$

7.1.3 Proteins (g)

The lower limit value of proteins that an elder should consume during a day represents 75% from his weight and the upper limit value of proteins that an elder should consume during a day is equal with the lower limit value of proteins multiplied by 110 %. The optimal value of proteins is expressed in grams and is computed as the average of these two limits.

$$Estimated\ Protein\ Value\ Lower\ Limit = weight \times 0.75 \quad (7.9)$$

$$Estimated\ Protein\ Value\ Upper\ Limit = 110\% \times Estimated\ Protein\ Value\ Lower\ Limit \quad (7.10)$$

$$Protein = \frac{Estimated\ Protein\ Value\ Lower\ Limit + Estimated\ Protein\ Value\ Upper\ Limit}{2} \quad (7.11)$$

7.1.4 Fats (g)

The value of fats that an elder should consume during the day varies between 20% and 35% from the optimal energy value divided by 900. The optimal value of fats is expressed in grams and is computed as the average of these two limits.

$$Estimated\ Fat\ Value\ Lower\ Limit = \frac{20 \times Energy}{900} \quad (7.12)$$

$$Estimated\ Fat\ Value\ Upper\ Limit = \frac{35 \times Energy}{900} \quad (7.13)$$

$$Fat = \frac{Estimated\ Fat\ Value\ Lower\ Limit + Estimated\ Fat\ Value\ Upper\ Limit}{2} \quad (7.14)$$

7.1.5 Vitamin A (ug)

There are two values for vitamin A that are evaluated: the average vitamin A value for men and the average vitamin A value for women. Also, there is a maximum value of vitamin A that an elder should consume.

$$Average\ Vitamin\ A\ Value\ (men) = 0.9 \quad (7.15)$$

$$Average\ Vitamin\ A\ Value\ (women) = 0.7 \quad (7.16)$$

$$Maximum\ Vitamin\ A\ Value = 3.0 \quad (7.17)$$

7.1.6 Vitamin B6 (mg)

There are two values for vitamin B6 that are evaluated: the average vitamin B6 value for men and the average vitamin B6 value for women. Also, there is a maximum value of vitamin B6 that an elder should consume.

$$Average\ Vitamin\ B6\ Value\ (men) = 1.7 \quad (7.18)$$

$$Average\ Vitamin\ B6\ Value\ (women) = 1.5 \tag{7.19}$$

$$Maximum\ Vitamin\ B6\ Value = 100.0 \tag{7.20}$$

7.1.7 Vitamin C (mg)

There are two values for vitamin C that are evaluated: the average vitamin C value for men and the average vitamin C value for women. Also, there is a maximum value of vitamin C that an elder should consume.

$$Average\ Vitamin\ C\ Value\ (men) = 90.0 \tag{7.21}$$

$$Average\ Vitamin\ C\ Value\ (women) = 75.0 \tag{7.22}$$

$$Maximum\ Vitamin\ C\ Value = 2000.0 \tag{7.23}$$

7.1.8 Vitamin D (ug)

The upper limit of vitamin D value that an elder should consume during a day is 110% multiplied by the lower limit of vitamin D value that an elder should consume during a day. The optimal value of vitamin D is the average of these two values.

$$Vitamin\ D\ Value\ Lower\ Limit = 10 \tag{7.24}$$

$$Vitamin\ D\ Value\ Upper\ Limit = 110\% \times 10 \tag{7.25}$$

$$VitaminD = \frac{Vitamin\ D\ Value\ Lower\ Limit + Vitamin\ D\ Value\ Upper\ Limit}{2} \tag{7.26}$$

7.1.9 Calcium (mg)

The lower limit value of calcium that an elder should consume during a day is 700 mg and the upper limit value of calcium that an elder should consume during a day is equal with the lower limit value of calcium multiplied by 110 %. The optimal value of calcium is expressed in milligrams and is computed as the average of these two limits.

$$Estimated\ Calcium\ Value\ Lower\ Limit = 700 \tag{7.27}$$

$$Estimated\ Calcium\ Value\ Upper\ Limit = 110\% \times Estimated\ Calcium\ Value\ Lower\ Limit \tag{7.28}$$

$$Calcium = \frac{Estimated\ Calcium\ Value\ Lower\ Limit + Estimated\ Calcium\ Value\ Upper\ Limit}{2} \tag{7.29}$$

7.1.10 Iron (mg)

For iron there are two values that are computed: the average iron value and the maximum iron value.

$$Average\ Iron\ Value\ (age < 50, men) = 8.0 \tag{7.30}$$

$$Average\ Iron\ Value\ (age < 50, women) = 18.0 \tag{7.31}$$

$$Average\ Iron\ Value\ (age \geq 50) = 8.0 \tag{7.32}$$

$$Maximum\ Iron\ Value\ = 45.0 \tag{7.33}$$

7.1.11 Sodium (mg)

For sodium there are also two values that are computed: the average sodium value and the maximum sodium value.

$$Average\ Sodium\ Value = 1500.0 \tag{7.34}$$

$$Maximum\ Sodium\ Value\ (age < 51) = 2300.0 \tag{7.35}$$

$$Maximum\ Sodium\ Value\ (age \geq 51) = 1500.0 \tag{7.36}$$

7.1.12 Saturated fat acids (g)

The two values that are computed for the saturated fat acids are the estimated saturated fat acids value upper limit and the optimal value of the saturated fat acids that an elder should consume during a day.

$$Estimated\ Sat\ Fat\ Acid\ Value\ Upper\ Limit = \frac{11 \times Estimated\ Energy\ Value}{900} \tag{7.37}$$

$$Saturated\ Fat\ Acid = \frac{Estimated\ Sat\ Fat\ Acid\ Value\ Upper\ Limit}{2} \tag{7.38}$$

7.1.13 Trans fat acids (g)

The two values that are computed for the trans fat acids are the estimated trans fat acids value upper limit and the optimal value of the trans fat acids that an elder should consume during a day.

$$Estimated\ Trans\ Fat\ Acid\ Value\ Upper\ Limit = \frac{Estimated\ Energy\ Value}{900} \tag{7.39}$$

$$Trans\ Fat\ Acid = \frac{Estimated\ Trans\ Fat\ Acid\ Value\ Upper\ Limit}{2} \tag{7.40}$$

7.1.14 Salt (mg)

The estimated salt value upper limit that an elder should consume during a day is 6 and the average value is equal with 6 divided by 2.

$$Estimated\ Salt\ Value\ Upper\ Limit = 6 \tag{7.41}$$

$$Salt = \frac{Estimated\ Salt\ Value\ Upper\ Limit}{2} \tag{7.42}$$

7.1.15 Alcohol (g)

The estimated alcohol value upper limit for men is 30 and the estimated alcohol value upper limit for women is 20.

$$Estimated\ Alcohol\ Value\ Upper\ Limit\ for\ Men = 30 \tag{7.43}$$

$$Estimated\ Alcohol\ Value\ Upper\ Limit\ for\ Women = 20 \tag{7.44}$$

$$Alcohol = \frac{Estimated\ Alcohol\ Value\ Upper\ Limit}{2} \tag{7.45}$$

7.1.16 Water (g)

The estimated water value lower limit is equal with 2000 and the estimated water value upper limit is equal with 2500.

$$Estimated\ Water\ Value\ Lower\ Limit = 2000 \tag{7.46}$$

$$Estimated\ Water\ Value\ Upper\ Limit = 2500 \tag{7.47}$$

$$Estimated\ Water\ Value = \frac{Estimated\ Water\ Value\ Lower\ Limit + Estimated\ Water\ Value\ Upper\ Limit}{2} \tag{7.48}$$

7.2 Assessing nutrition related diagnostics

This section describes the equations used for the determination of the energy intake and the fluid intake related diagnostics.

7.2.1 Energy Intake Diagnostics

There are three types of energy intake diagnostics: excessive energy intake, suboptimal energy intake and normal energy intake.

$$\begin{aligned}&Excessive\ Energy\ Intake:\\&\qquad Consumed\ Energy\ Value > Estimated\ Energy\ Value\ Upper\ Limit\end{aligned} \tag{7.49}$$

$$\begin{aligned}&Suboptimal\ Energy\ Intake:\\&\qquad Consumed\ Energy\ Value < Estimated\ Energy\ Value\ Lower\ Limit\end{aligned} \tag{7.50}$$

$$\begin{aligned}&Normal\ Energy\ Intake:\\&\qquad Estimated\ Energy\ Value\ Upper\ Limit \geq Consumed\ Energy\ Value\\&\qquad and\\&\qquad Consumed\ Energy\ Value \geq Estimated\ Energy\ Value\ Lower\ Limit\end{aligned} \tag{7.51}$$

7.2.2 Fluid Intake Diagnostics

There are three types of fluid intake diagnostics: excessive fluid intake, suboptimal fluid intake and normal fluid intake.

$$\begin{aligned}&Excessive\ Fluid\ Intake:\\&\qquad Consumed\ Water\ Value > Estimated\ Water\ Value\ Upper\ Limit\end{aligned} \tag{7.52}$$

$$\begin{aligned}&Suboptimal\ Fluid\ Intake:\\&\qquad Consumed\ Water\ Value < Estimated\ Water\ Value\ Lower\ Limit\end{aligned} \tag{7.53}$$

$$\begin{aligned}
&Normal\ Fluid\ Intake: \\
&\qquad Estimated\ Water\ Value\ Upper\ Limit \geq Consumed\ Water\ Value \\
&\qquad and \\
&\qquad Consumed\ Water\ Value \geq Estimated\ Water\ Value\ Lower\ Limit
\end{aligned} \tag{7.54}$$

7.3 Assessing weight related diagnostics

In order to assess the weight related diagnostics it is necessary to compute the value of the Body Mass Index (BMI) first.

7.3.1 Body Mass Index

The value of the Body Mass Index (BMI) depends on the height and on the weight of the elders.

$$BMI = \frac{weight(Kg)}{height(m) \times height(m)} \tag{7.55}$$

7.3.2 Weight Diagnostics

There are four types of weight diagnostics: underweight, normal weight, overweight and obesity.

$$Obesity : BMI \geq 30 \tag{7.56}$$

$$Overweight : 30 > BMI \geq 25 \tag{7.57}$$

$$Normal\ weight : 25 > BMI \geq 18.5 \tag{7.58}$$

$$Underweight : 18.5 > BMI \tag{7.59}$$

7.4 Computation of the score for a custom defined diet

This section describes the questionnaire used for the evaluation of the score for a custom defined Mediterranean Diet.

Table 7.1: Mediterranean Diet Adherence Questionnaire

NR	Question	Implementation Details
1	Do you use olive oil as main culinary fat?	The value of olive oil consumed during the last day is > 0 grams.

2	Do you consume more than 4 tablespoons of olive oil per day? (1 tablespoon = 14.3 grams)	The value of olive oil consumed during the last day is > 57.2 grams.
3	Do you consume more than 2 servings of vegetables per day? (1 serving = 200 g)	The total value of vegetables consumed during the last day is > 400.0 grams.
4	Do you consume more that 3 fruit units (including natural fruit juices) per day?	The number of fruit units consumed during the last day is > 3.
5	Do you consume less than one serving of red meat, hamburger or meat products (ham, sausage, etc.) per day? (1 serving = 150 grams)	The value of meat consumed during the last day is < 150.0 grams.
6	Do you consume less than one serving of butter, margarine or cream per day? (1 serving = 12 grams)	The value of fats consumed during the last day is < 12 grams.
7	Do you drink less than one sweet or carbonated beverage per day?	The number of sweet beverages consumed during the last day is < 1.
8	Do you drink wine everyday?	The value of wine consumed in each of the last 7 days is > 0.
9	Do you consume more than 3 servings of legumes per week? (1 serving = 150 grams)	The total value of legumes consumed in the last seven days is > 450.0 grams.
10	Do you consume more than 3 servings of fish per week? (1 serving = 150 grams)	The total value of fish consumed in the last 7 days is > 450.0 grams
11	Do you consume commercial sweets or pastries (not homemade), such as cakes, cookies, biscuits or custard less than 3 times per week?	The number of foods which contain pastries consumed during the last day is < 3.
12	Do you consume more than 3 servings of nuts (including peanuts) per week? (1 serving = 30 grams)	The total value of nuts consumed in the last seven days is > 90 grams.
13	Do you preferentially consume chicken, turkey or rabbit meat instead of veal, pork, hamburger or sausage?	The total value of birds meat consumed in the last seven days is > the total value of veal value consumed in the last seven days.
14	Do you consume pasta and rice at least 2 times per week?	The number of foods which contain pasta or rice consumed in the last 7 days is > 1.

The elements from the ontology that are considered in the evaluation of the score of an elder for a Mediterranean Diet can be seen in Figure 7.1.

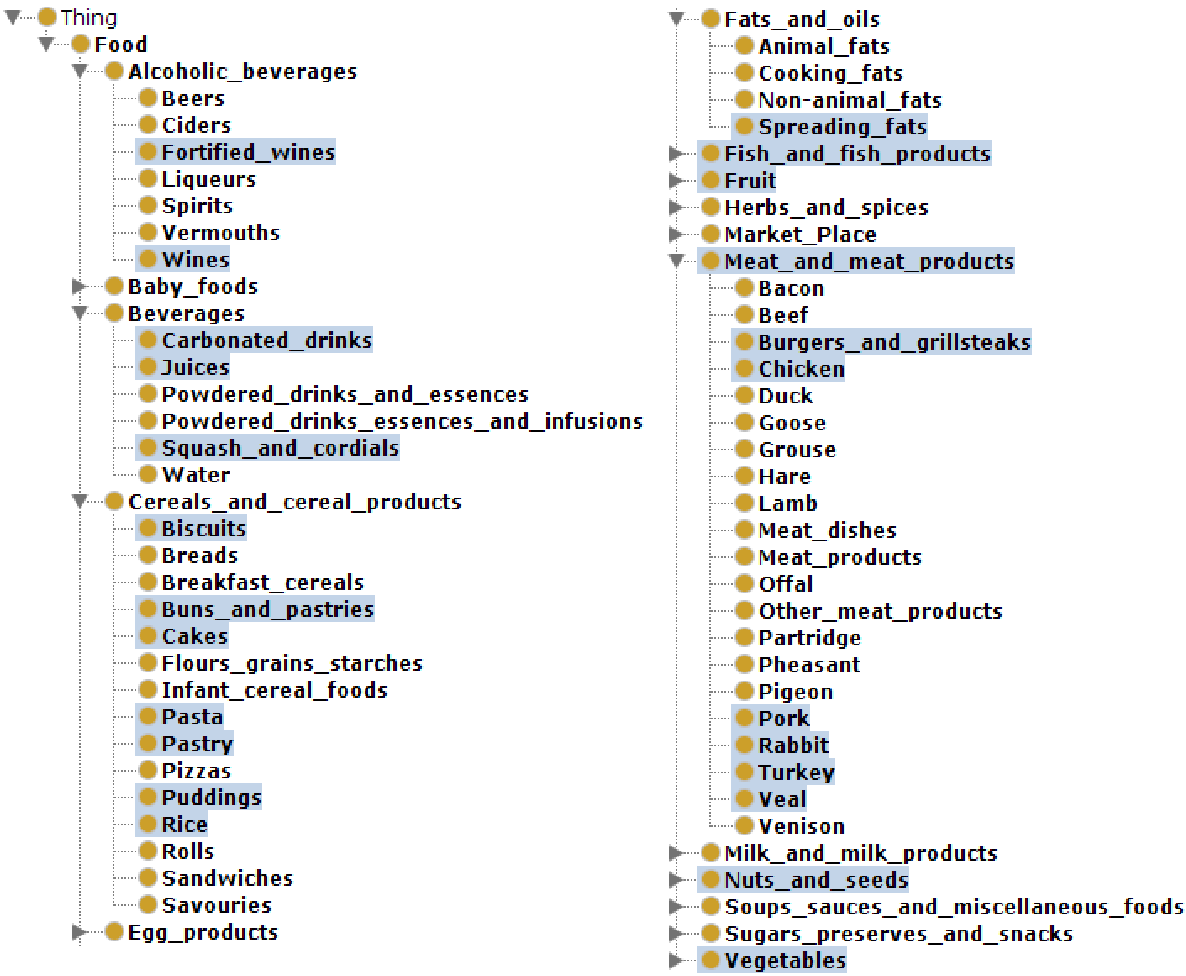

Figure 7.1: Conceptual architecture of the system

7.5 Mapping tools between ontologies and databases

This section describes four possible approaches [120] that can be used in order map the information from the database to ontologies. The mapping between the databases and the ontologies is necessary because the information about the elders such as height, weight, PAF, and so on are stored in the database, while the reasoning rules that are used to represented the nutritional recommendations are written in the ontologies. Each subsection describes the approach used to map a database to an ontology.

7.5.1 KAON2

The KAON2 tool can be used to manipulate ontologies written in Web Ontology Language - Description Logic (OWL-DL) and the information written in the ontologies can be queried using SPARQL. Some of the drawbacks of KAON2 are represented by the fact that it can not handle enumerated classes and very large numbers. A detailed architecture of KAON2 is described in Figure 7.2.

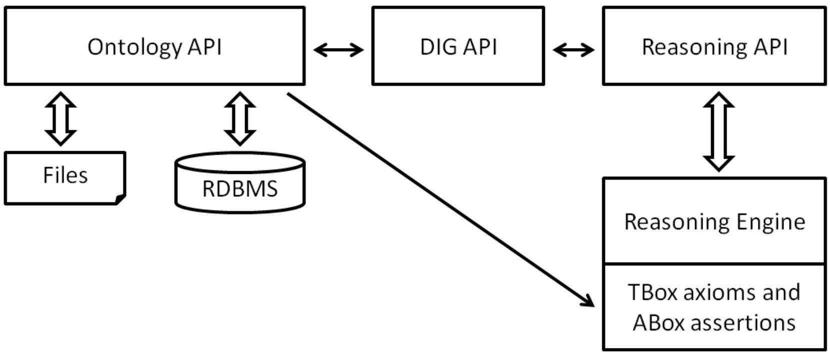

Figure 7.2: KAON2 Architecture (adapted from [1])

The manipulation of the ontology can be done using the *Ontology API* module. The ontologies can be loaded either in the OWL RDF format or in the OWL XML format. The assertions about the individuals (ABox assertions) can be persisted in a relational database (RDBMS). The entities of the ontology can be mapped to database tables and the database can be queried on the fly during reasoning. A variety of reasoning tasks can be invoked using the *Reasoning API* module which communicates with the *Reasoning Engine* module. Also, these APIs can be invoked through the *DL Implementors Group (DIG) API* remotely.

7.5.2 Jena and D2RQ

The architecture presented in Figure 7.3 describes how a relational database can be connected to an ontology. The mapping between the relational *database* and the ontology is realized through a mapping file that has the extension *.ttl*. Using this mapping file, a *D2RQ Data* model is obtained. This model contains the individuals that will populate the ontology. The ontologies can be written in several *.owl files* and they can be organized in a *.rdf file* which also manages the ontology imports. From these files the *Ontology Data* is obtained, which contains the description of the ontology. Using the *Pellet Reasoner* or any other reasoner an *Inferred Model* is generated from which the *Ontology Model* is obtained. The TBox component is represented by the *Ontology Model* while the ABox component is represented by the *D2RQ Data*. The model obtained after the combination of these two sources can be queried by using SPARQL queries.

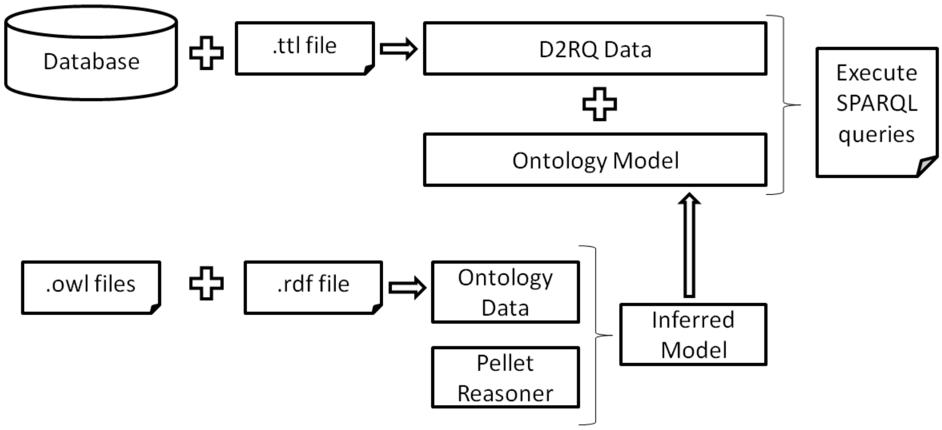

Figure 7.3: An architecture that uses D2RQ and Jena (adapted from [2])

7.5.3 Jena and R2RML

An architecture that uses Jena and R2RML is presented in Figure 7.4. The input is represented by a *Source database* and the output is represented by an *RDF Graph*. The *R2RML Parser* is composed from three parts: the *Mapping file*, the *Parser* and the *Generator*. The *Mapping file* is used to map the relational database to result sets. Next, the *Parser* is used in order to generate a set of instructions based on the mapping file. Finally, the *Generator* is used to instantiate an *RDF Graph*.

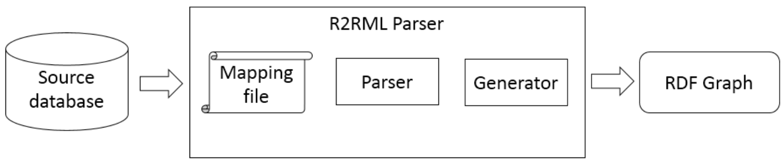

Figure 7.4: An architecture that uses R2RML and Jena (adapted from [3])

7.5.4 OWL API and SQL

An approach that uses OWL API and SQL is presented in Figure 7.5. An *.owl file* is used to store the ontology concepts. The data that is used to populate the database is persisted in a relational database. The *RDB to RDF translator* module is used convert the information that is stored in the database into information that has an *RDF* format. The model which is obtained after populating the ontology with information that comes from the database can be queried by a reasoner.

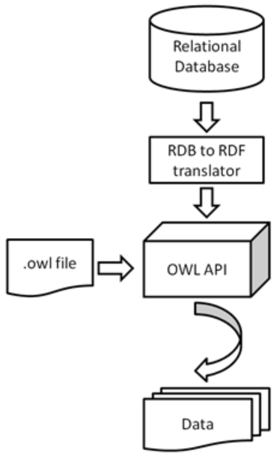

Figure 7.5: An architecture that uses OWL API

7.6 Use case evaluation

This section describes how the mapping between the ontologies and the databases is achieved using the four approaches described before. The ontology will have a set of reasoning rules and thus a significant amount of time will be dedicated to the reasoning process. The two parameters that are compared in Section 9 are the time necessary to load the instances from the database in the ontology and the time required to query the information from the ontology using SPARQL queries. The ontology that is used in order to show how the four approaches work is a small part of the Diets Ontology and is described in Figure 7.6.

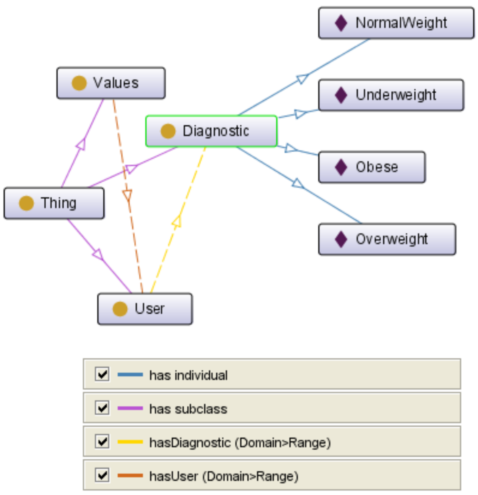

Figure 7.6: Diagnostic Ontology

The three main components of the ontology are the people, the values and the diagnostics. The people correspond to the *User Ontology Class*, the values correspond to the *Values Ontology Class* and the diagnostics correspond to the *Diagnostic Ontology Class*. There are four types of diagnostics: *Overweight*, *Obese*, *Underweight* and *Normal Weight*. Each *User* is associated with the next information: *firstName*, *lastName*, *gender* and *age*. The object property that links a *User* to a *Diagnostic* is the *hasDiagnostic* object property. The *User* and the *Values* are linked by the *hasUser* object property. The *Values Ontology Class* has the following three datatype properties: *hasWeight*, *hasHeight* and *hasBMI*. The *Diagnostic Ontology* has reasoning rules for the determination of the next values: *BMI*, *Underweight*, *NormalWeight*, *Overweight* and *Obese*.

$$BMI = \frac{Weight(Kg) \times 10000}{Height(cm)^2} \tag{7.60}$$

$$BMI < 18.5 \Rightarrow Underweight \tag{7.61}$$

$$18.5 \leq BMI < 25 \Rightarrow NormalWeight \tag{7.62}$$

$$25 \leq BMI < 30 \Rightarrow Overweight \tag{7.63}$$

$$30 \leq BMI \Rightarrow Obese \tag{7.64}$$

The database used for the ontology mapping consists of two tables: the *users* table and the *values* table. The table *users* contains information about the *firstName*, the *lastName*, the *gender* and the *age* of the elder. The table *values* contains information about the *weight* and the *height* of the elder. The Figure 7.7 describes the structure of the database used for the use case evaluation of the four approaches.

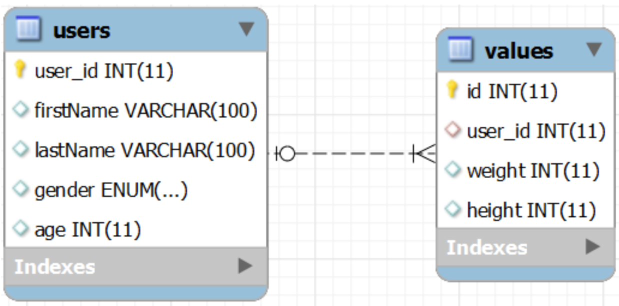

Figure 7.7: Database model

7.6.1 KAON2 evaluation

In the case of using KAON2 for the mapping between the ontology and the database, an *.xml file* is used. This file describes how the database concepts are mapped to ontology concepts. The first step is to describe the prefixes of the ontology, the second step is to specify how the connection to the database is realized and the third step is to map the database tables to ontology classes. The ontology classes are mapped to database tables, the columns are mapped to datatype properties, the foreign keys are mapped to object properties and the ontology individuals are mapped to rows from the database tables. An illustrative example of mapping between an ontology class and a database table using KAON2 can be seen in Listing 7.1.

Listing 7.1: Map an Ontology Class using KAON2

```
<db:OWLClass db:name="http://www.semanticweb.org/ontologies/2015/2/elders#User">
    <db:Table db:tableName="users">
        <db:String db:fieldName="user_id"
                db:uriPrefix="http://www.semanticweb.org/ontologies/2015/2/elders#User_"
                db:primaryKey="true"/>
    </db:Table>
</db:OWLClass>
```

The reasoning rules are written dynamically in code and then they are added to the ontology. An example of reasoning rule can be seen in Listing 7.2. The rule described in this listing specifies that if the *Body Mass Index* of an elder is in the interval *[18.5, 25.0)* then the diagnostic that corresponds to him is *NormalWeight*.

Listing 7.2: KAON2 Reasoning Rule for Underweight Diagnostic

```
Rule rule_normalWeight = KAON2Manager.factory().rule(
KAON2Manager.factory().literal(true, hasDiagnostic,
                new Term[] { U, normalWeight }),

new Literal[] {
                hasUser_M_U,
                KAON2Manager.factory().literal(true, bmi,
                                        new Term[] { M, BMI }),
                KAON2Manager.factory().literal(
                        true,
                        KAON2Manager.factory().ifTrue(2),
                        new Term[] {
                                KAON2Manager.factory().constant(
                                        "$1␣>=␣18.5"), BMI }),
                KAON2Manager.factory().literal(
                        true,
                        KAON2Manager.factory().ifTrue(2),
                        new Term[] {
                                KAON2Manager.factory().constant(
                                        "$1␣<␣25.0"), BMI }) });
```

The ontology can be queried by using SPARQL queries. An example of SPARQL query that returns the first name, the last name and the diagnostic of an elder is described in Listing 7.3.

Listing 7.3: KAON2 SPARQL Query for Retrieval of Diagnostics

```
public static final String SPARQL = "SELECT␣?y␣?z␣?d␣WHERE␣{␣"
        + "?x␣rdf:type␣<http://www.semanticweb.org/ontologies/2015/2/elders#User>␣;␣"
        + "<http://www.semanticweb.org/ontologies/2015/2/elders#hasFirstName>␣?y␣;␣"
        + "<http://www.semanticweb.org/ontologies/2015/2/elders#hasLastName>␣?z␣.␣"
        + "?x␣<http://www.semanticweb.org/ontologies/2015/2/elders#hasDiagnostic>␣?d␣.␣"
        + "␣}";
```

7.6.2 Jena and D2RQ evaluation

The process of mapping between the ontology and the database when Jena and D2RQ are used is similar with the process used in the case of KAON2. The mapping is realized using a *.ttl file*. The mapping file starts with the declaration of the prefixes used by the ontology that is mapped, continues with the declaration of the parameters necessary for the connection to the database and ends with the mapping of each table, column and foreign key to an ontology class, an ontology datatype property and an ontology object property, respectively. Listing 7.4 shows how to map a datatype property using the D2RQ tool.

Listing 7.4: Map a Datatype Property using D2RQ

```
# Data Property weight
map:weight a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:values;
    d2rq:property elders:hasWeight;
    d2rq:column "values.weight";
    d2rq:datatype xsd:int;
        .
```

The rules were written in the Semantic Web Rule Ontology Language (SWRL) using the Protege tool. An illustrative example of how these rules can be represented in SWRL can be seen in Listing 7.5.

Listing 7.5: Reasoning Rules Written in SWRL

```
Values(?x), hasHeight(?x, ?h), hasWeight(?x, ?w), divide(?r, ?w, ?n),
        multiply(?n, ?h, ?h), multiply(?rez, ?r, 10000) -> hasBMI(?x, ?rez)
User(?x), Values(?y), hasUser(?y, ?x), hasBMI(?y, ?b),
        greaterThanOrEqual(?b, 30.0) -> hasDiagnostic(?x, Obese)
User(?x), Values(?y), hasUser(?y, ?x), hasBMI(?y, ?b),
        greaterThanOrEqual(?b, 25.0), lessThan(?b, 30.0)
        -> hasDiagnostic(?x, Overweight)
User(?x), Values(?y), hasUser(?y, ?x), hasBMI(?y, ?b),
        greaterThanOrEqual(?b, 18.5), lessThan(?b, 25.0)
        -> hasDiagnostic(?x, NormalWeight)
User(?x), Values(?y), hasUser(?y, ?x), hasBMI(?y, ?b),
        lessThan(?b, 18.5) -> hasDiagnostic(?x, Underweight)
```

The data can be queried by using SPARQL queries. Also, there is the possibility to use different reasoners such as: the *Pellet Reasoner*, the *RDF Reasoner* and the *Generic Reasoner*. An illustrative example is presented in Listing 7.6.

Listing 7.6: SPARQL for the Retrieval of Diagnostics

```
SELECT ?firstName ?lastName ?diagnostic WHERE {
        ?user rdf:type elders:User . ?user elders:hasFirstName ?firstName .
        ?user elders:hasLastName ?lastName . ?user elders:hasDiagnostic ?diagnostic .
        }
```

7.6.3 Jena and R2RML evaluation

The approach that uses Jena and R2RML for the mapping between the ontology and the database also uses a mapping *.ttl file*. One difference between D2RQ and R2RML is represented by the fact that the parameters that are necessary for the connection to the database are not written in the *.ttl file* in the case of R2RML. An illustrative example of a table that is mapped to an ontology class is presented in Listing 7.7.

Listing 7.7: Map a Datatype Property using R2RML

```
rr:predicateObjectMap
[
    rr:predicate        elders:hasAge ;
    rr:objectMap     [ rr:column "\"age\""; ]
]
```

An alternative for using the SWRL to represent the reasoning rules is to use Jena rules. Listing 7.8 presents a Jena rule.

Listing 7.8: Jena Rule for the Determination of the NormalWeight Diagnostic

```
[NormalWeightRule:
  (?x rdf:type :User),
  (?y rdf:type :Measurements),
  (?y :hasUser ?x),
  (?y :hasBMI ?b),
  ge(?b, 18.5),
  lessThan(?b, 25.0),
->
  (?x :hasDiagnostic :NormalWeight),
]
```

The information can be queried using SPARQL queries in a way similar with that presented in the previous subsection.

7.6.4 OWL API and SQL evaluation

In the case of OWL API and SQL the approach is different. The information from the database is retrieved using *sql* queries and then it is converted into ontology information which is used to populate the ontology. A table from the database corresponds to an ontology class, the rows of the tables are individuals that populate the ontology, the foreign keys are mapped to object properties and the columns that are not associated with foreign keys are mapped to datatype properties. OWL API does not provide SPARQL support as Jena. Instead of using SPARQL queries to retrieve the information which is generated after reasoning, the alternative is to call the reasoner. Another approach is to add the OWL API ontology model to a Jena ontology model and to query the Jena model with SPARQL.

Chapter 8

Diets generation module

This chapter presents four approaches for the generating of healthy meals recommendations for the elders. The chapter is organized in four parts: the first part describes the problem that is solved, the second part presents the Particle Swarm Optimization (PSO) Algorithm, the third part describes the Glow-worm Swarm Optimization Algorithm and the fourth part presents the Spark concepts that are used to improve the PSO and the GSO.

8.1 Definition of the problem

The problem that the *Diets generation module* must solve is the following one: *Given a repository of food packages from different food providers, find an optimal combination of food packages for each part of the day - breakfast, brunch, lunch, snack, dinner - that matches the profile of an older adult (e.g. age, weight, height, physical activity factor), the preferences of the older adult (e.g. price, preparation time, reliability preferences) and the nutritionist's recommended diet.*

8.1.1 Solution representation

The representation of the solution and the PSO algorithm are adapted after the ones described in [121] for the context associated with the problem described in this thesis. The solution is defined as a set of food packages, one for each part of the day - breakfast, brunch, lunch, snack and dinner. A description of the solution can be seen in Figure 8.1. A food packet contains the following information: *(a) Nutritional components* - numerical values for eleven nutrients: energy (kcal), carbohydrates (g), proteins (g), fats (g), vitamin A (ug), vitamin B6 (mg), vitamin C (mg), vitamin D (ug), calcium (mg), iron (mg), sodium (mg), *(b) Preparation time* - the time required to prepare the packet, *(c) Price* - the price of the packet, *(d) Reliability dimensions* - there are four reliability dimensions that are measured: size, aspect, taste, smell, *(e) Diets* - there can be several types of diets such as vegetarian, low sodium and low fat, *(f) Part of the day* - the part of the day associated with the food packet can be one of the following: breakfast, brunch, lunch, snack, dinner.

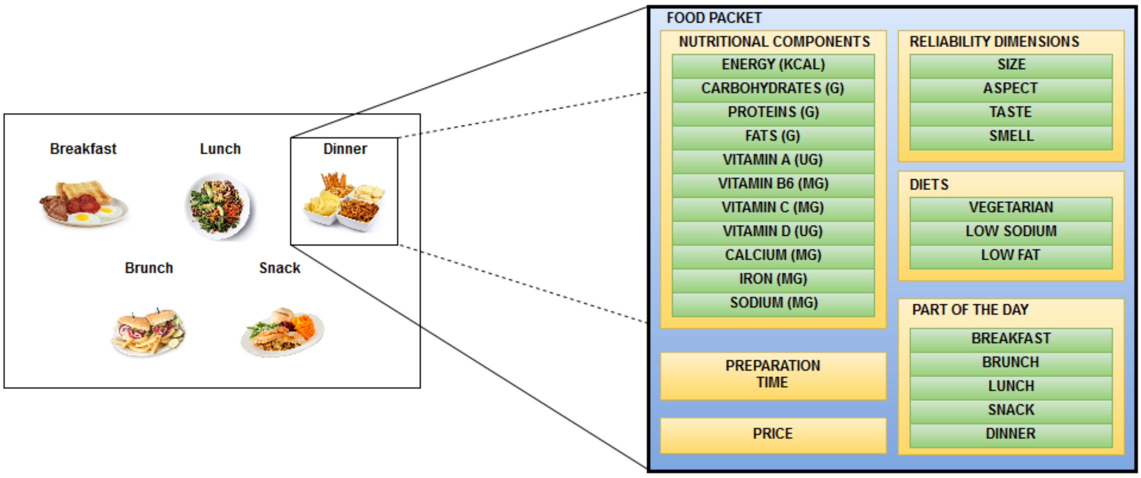

Figure 8.1: Solution representation

8.1.2 Solution evaluation

The solution is evaluated based on the following four criteria: the nutrient component, the price component, the time component and the reliability component.

Total fitness

The fitness for the evaluation of a solution is the following:

$$F_{total}(sol) = \frac{1}{w_{nutrient} + w_{price} + w_{time} + w_{reliability}} \times (w_{nutrient} \times F_{nutrient}(sol) + w_{price} \times F_{price}(sol) + w_{time} \times F_{time}(sol) + w_{reliability} \times F_{reliability}(sol)) \quad (8.1)$$

where $F_{total}(sol)$ is the total fitness of the solution, $F_{nutrient}(sol)$ is the evaluation of the solution from the nutrients intake perspective, $F_{price}(sol)$ is the evaluation of the solution from the price perspective, $F_{time}(sol)$ is the evaluation of the solution from the time perspective, $F_{reliability}(sol)$ is the evaluation of the solution from the reliability perspective and $w_{nutrient}, w_{price}, w_{time}, w_{reliability}$ are weights associated with the four fitness functions. The four fitness function components and their weights are described next.

Nutrition component fitness

The *nutrition component fitness* evaluates the nutrition component by comparing it to the ideal value. The two types of information that are required are the following: *the quantity of the component of the solution which is evaluated* (e.g. if one meal is associated with 100 g proteins then the quantity of proteins in the solution is equal with 100) and *the optimal quantity of the component* as it is inferred from the recommendations proposed by the nutritionist (e.g. for an older adult the ideal value that should be consumed is 80). The solutions that are closer to the ideal values have a score equal with 0 while the solutions which are farther from the ideal values have greater values. The total nutrition component fitness has the following formula:

$$F_{nutrient}(sol) = \sum_{i=0}^{10} p_i \times \left| \frac{nutrient\ ideal\ value_i - nutrient\ value_i}{nutrient\ ideal\ value_i} \right| \tag{8.2}$$

where

$$\sum_{i=0}^{10} p_i = 1 \tag{8.3}$$

The values $nutrient\ ideal\ value_i$ with $i = \overline{0, 10}$ represent the ideal values for energy, carbohydrates, proteins, fats, vitamin A, vitamin B6, vitamin C, vitamin D, calcium, iron and sodium. If one term of the form $p_i \times \left| \frac{nutrient\ ideal\ value_i - nutrient\ value_i}{nutrient\ ideal\ value_i} \right|$ is greater than 0.5 then the value of this term will be equal with 2. In this way the solutions which are not close to the ideal values have greater scores.

Price component fitness

As in the case of the *nutrition component fitness*, the *price component fitness* is evaluated based on the total price of the solution and the ideal price as it is specified by the elder. In the case this value is greater than 0.5 for any of the parts that compose the recommendation (e.g. for breakfast) then this value is replaced by 2 for that component.

$$F_{price}(sol) = \left| \frac{price\ ideal\ value - price\ actual\ value}{price\ ideal\ value} \right| \tag{8.4}$$

Time component fitness

The time component fitness is evaluated based on the total time which is necessary to prepare all the foods of the solution and the ideal preparation time as it is specified by the elder. In the case this value is greater than 0.5 for any of the parts that compose the recommendation (e.g. for lunch) then this value is replaced by 2 for that component.

$$F_{time}(sol) = \left| \frac{preparation\ time\ ideal\ value - preparation\ time\ actual\ value}{preparation\ time\ ideal\ value} \right| \quad (8.5)$$

Reliability component fitness

There are four reliability dimensions: taste, smell, size and aspect. The formula is similar with the one used for the computation of the *nutrition component fitness*.

$$F_{reliability}(sol) = \sum_{i=0}^{3} q_i \times \left| \frac{rd\ ideal\ value_i - rd\ value_i}{rd\ ideal\ value_i} \right| \quad (8.6)$$

where

$$\sum_{i=0}^{3} q_i = 1 \quad (8.7)$$

The $rd\ ideal\ values_i$ with $i = \overline{0,3}$ describe the ideal reliability dimensions preferences for taste, smell, size and price as they are specified by the elder and the $rd\ values_i$ with $i = \overline{0,3}$ describe the actual values of these dimensions as they appear in the solution.

Weights

The importance of a component in the calculation of the value of the fitness function is represented by the weight which is associated with it. Each of the four components of the fitness function (e.g. nutrition, price, time, reliability) is associated with a weight which reflects the preferences of the elders for specific parts of the solution and the preferences of the nutritionists for specific nutrients.

8.2 Generating of Diets using PSO

The Particle Swarm Optimization algorithm used in this thesis is a modification of the classical Particle Swarm Optimization algorithm according to the domain of generating healthy meals recommendations for elders. The algorithm is briefly described below in

Algorithm 1.

Data: foodOffers, personalProfile, dietaryRecommendation, swarmSize, numberOfIterations, a, b, c

Result: optimalElderDailyMenu

optimalElderDailyMenu = null;
t = 0;
swarm = GenerateParticlesRandomly(swarmSize, foodOffers, personalProfile, dietaryRecommendation);
$position_{global-best}$ = GetGlobalBest(swarm, foodOffers, personalProfile, dietaryRecommendation);
for $i = 1 \rightarrow swarmSize$ **do**
    $position_{local-best}(i)$ = GetLocalBest(swarm, foodOffers, personalProfile, dietaryRecommendation);
end
while $t < numberOfIterations$ **do**
    for $i = 1 \rightarrow swarmSize$ **do**
        $velocity^{t+1}(i) = a \times velocity^{t}(i) + b \times |position_{local-best}(i) - position^{t}(i)| + c \times |position_{global-best} - position^{t}(i)|$;
        $position^{t+1}(i) = GeneratePosition(position^{t}(i), velocity^{t+1}(i), foodOffers, personalProfile, dietaryRecommendation$;
        $position_{local-best}(i) = UpdateLocalBest(position^{t+1}(i), position_{local-best}(i))$;
        $position_{global-best} = UpdateGlobalBest(position_{global-best}, position_{local-best}(i))$;
    end
    t = t + 1;
end
optimalElderDailyMenu = $GetBestParticle(swarm, position_{global-best})$

Algorithm 1: PSO Algorithm

The first steps of the algorithm initialize the particles randomly in the search space. From the set of the generated particles, the position which is the global best is chosen and for each particle the local best position encountered so far is equal with the current position. While the stopping condition is not true (e.g. the number of iterations is smaller than the threshold), for each particle from the swarm the velocity and the position are updated. After the velocity and the position are updated, the local best position of the particle is updated and the global best position takes a new value if the new local best position of the current particle is better. Finally, the algorithm returns the recommendation associated with the particle from the swarm that corresponds to the best global position.

A mapping between the PSO algorithm and the proposed optimization problem can be seen in Table 8.1.

Table 8.1: Mapping between PSO and the healthy meals generating problem

PSO Concept	Healthy meals generating problem concept
position	a vector of eleven elements, one for each nutrient
velocity	a vector of eleven elements, one for each nutrient
particle	a combination of five meals; it has a position and a velocity
swarm	a set of particles
search space	the set of all possible combinations of five meals (e.g. breakfast, brunch, lunch, snack, dinner)

8.3 Generating of Diets using GSO

The Glow-worm Swarm Optimization algorithm used in this thesis is a modification of the classical version of the GSO algorithm [122] according to the domain of the problem of generating healthy meals recommendations for elders. The algorithm is presented in

Algorithm 2.

Data: foodOffers, personalProfile, dietaryRecommendation, swarmSize, numberOfIterations, luciferinInitialValue, luciferinDecayConstant, luciferinEnhancementConstant, BETA, stepSize, numberOfNeighbors, neighborhoodRangeDecision, sensorRange

Result: optimalElderDailyMenu

optimalElderDailyMenu = null;
t = 0;
glowworms = GenerateGlowWormsRandomly(swarmSize, foodOffers, personalProfile, dietaryRecommendation);
for $i = 1 \rightarrow swarmSize$ **do**
  $luciferin_i^0 = luciferinInitialValue$;
  $neighborhoodRange_i^0 = neighborhoodRangeDecision$;
end
while $t <$ *numberOfIterations* **do**
  for $i = 1 \rightarrow swarmSize$ **do**
    $luciferin_i^{t+1} = (1 - luciferinDecayConstant) \times luciferin_i^t + luciferinEnhancementConstant \times F_{total}(glowworms_i)$
  end
  for $i = 1 \rightarrow swarmSize$ **do**
    N_i^t = find neighbors of the $glowworms_i$;
    for *each glow-worm j in* $neighbors_i^t$ **do**
      $P_{ij} = \frac{luciferin_j^t - luciferin_i^t}{\sum_{k \in N_i^t} luciferin_k^t - luciferin_i^t}$;
    end
    select glow-worm j using P_{ij};
    update the position of the $glowworms_i$ with the equation $glowworms_i^{t+1} = glowworms_i^t + stepSize \times \frac{glowworms_j^t - glowworms_i^t}{||glowworms_j^t - glowworms_i^t||}$;
    update the decition range of the $glowworms_i$ with the equation $neighborhoodRange_i^{t+1} = min(sensorRange, max(0, neighborhoodRange_i^t + BETA \times (numberOfNeighbors - |N_i^t|)))$;
  end
  t = t + 1;
end
optimalElderDailyMenu = $GetBestGlowWorm(swarm)$

Algorithm 2: GSO Algorithm

The algorithm starts with the random generation of the initial population of glow-worms. For each glow-worm from the swarm the luciferin value and the neighborhood range are initialized with some constant values. While the number of iterations is less than a specified threshold, the following steps are executed. In the luciferin update phase, for each glow-worm from the swarm the luciferin value is updated. In the movement phase, for each glow-worm i a neighbor j with greater luciferin value is selected such that the probability of selecting j is proportional with the luciferin value of the j. Then the position and the neighborhood of the glow-worm i are updated based on the previous positions of the glow-worms i and j. After the stopping condition is met, the algorithm returns the best glow-worm (e.g. the best combination of five meals).

A mapping between the GSO algorithm and the proposed optimization problem can be seen in Table 8.2.

Table 8.2: Mapping between GSO and the healthy meals generating problem

GSO Concept	**Healthy meals generating problem concept**
position	a vector of eleven elements, one for each nutrient
luciferin	a value proportional with the fitness of a recommendation
glow-worm	a combination of five meals; it has a position and a luciferin value
swarm	a set of glow-worms
search space	the set of all possible combinations of five meals (e.g. breakfast, brunch, lunch, snack, dinner)

8.4 Generating of Diets using PSOS and GSOS

The PSOS and the GSOS algorithms are built on the PSO and GSO algorithms presented in the previous sections but in which some instructions are replaced by Spark methods. The retrieval of the information from the database is done only once at the beginning of the algorithm and it is stored in Spark specific data structures. The explanation of the main concepts of Spark used in this context is presented in Table 8.3.

Table 8.3: Spark concepts used to improve the PSO and the GSO algorithms

Spark Concept	**Healthy meals generating problem concept**
RDD	there are five RDDs: breakfastRDD, brunchRDD, lunchRDD, snackRDD, dinnerRDD
filter	filter the packets that are not compatible with a specific diet associated with the elder
map	convert the information retrieved from the relational database in the form of Java objects
data frame	data structure similar with the database tables which contains information from multiple tables

cache	store the RDDs and the data frame in memory in order to access them faster
collect	collect the information obtained after the mapping of filtering operations

Chapter 9

Experimental Results

This section is organized in two subsections. The first subsection presents the experimental results that were obtained after the comparison of the four mapping approaches between the ontologies and the databases: KAON2, Jena and D2RQ, Jena and R2RML and OWL API and SQL. The second part describes the experimental results obtained after using the four algorithms: Particle Swarm Optimization (PSO), Particle Swarm Optimization with Spark (PSOS), Glow-worm Swarm Optimization (GSO) and Glow-worm Swarm Optimization with Spark (GSOS).

9.1 Reasoning Module Evaluation

The four approaches described in Section 7 were tested by considering *10* test cases. Each test case is associated with a number of users that are stored in the database. The number of users takes values from the set { *100, 200, 300, 400, 500, 600, 700, 800, 900, 1000* }. The users are generated randomly in the database together with their measurements. The height of the users takes values from *165* to *184* and the weight of the users takes values from *50* to *100*. As output, in each of the four cases the information about users that is retrieved is the following: the first name, the last name and the diagnostic. There are two numerical values that are compared: the time required to load the information from the database in the ontology and the time required to retrieve the information about the users. Figure 9.1 presents the time required to load the information from the relational database in the ontology and Figure 9.2 presents the time required to retrieve the diagnostics of the users after the reasoning operations are performed.

9.2 Diets Generation Module Evaluation

In order to evaluate the performance of the four proposed approaches, a machine with the following parameters was used: Intel(R) Core(TM) i7-4500 CPU @ 1.80 GHz 2.40 GHz, 8.00 GB RAM, 64-bit Operating System, Windows 7. The search space consist

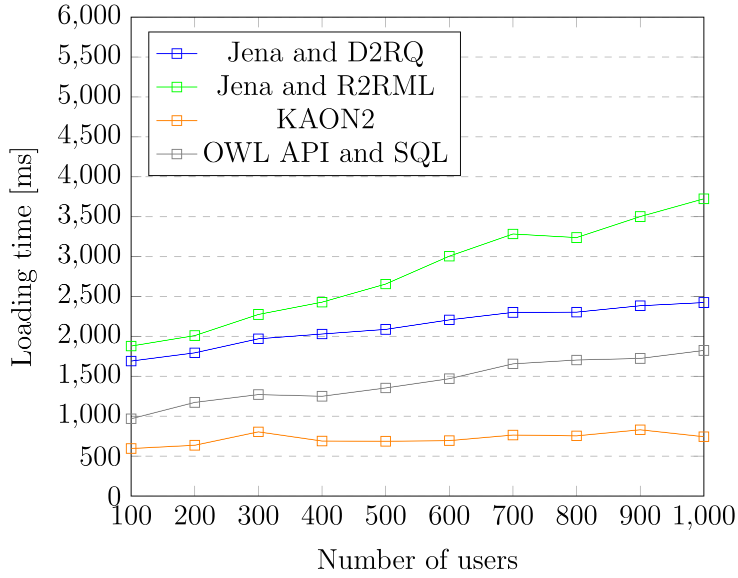

Figure 9.1: Loading time in milliseconds

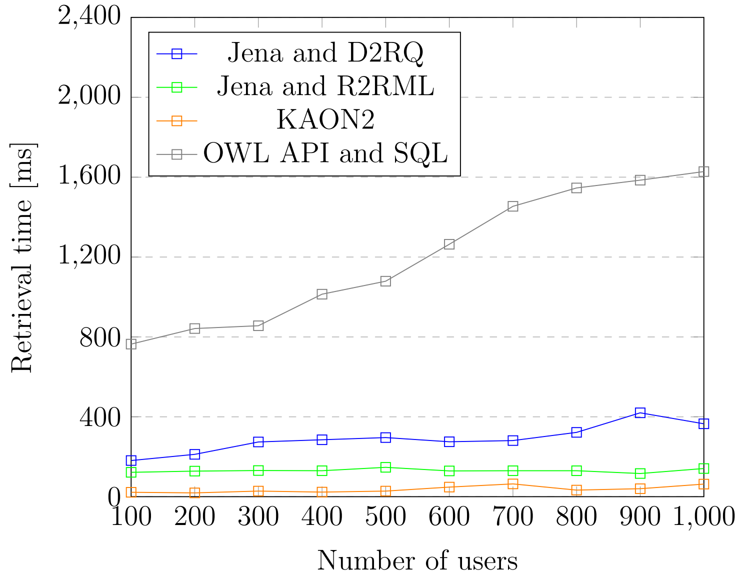

Figure 9.2: Retrieval time in milliseconds

of around 2500 packages. Different profiles of users were tested, but the thesis shows the results obtained only for a user with the following information: man, 70 years old, weight 79 Kg, height 180 cm, price preference of 40 euro, and preparation time preference of 30 minutes. His preferences are, in decreasing order of importance, the following: price, time and reliability. The preferences for the reliability dimensions are, in decreasing order of importance, the following: taste, aspect, smell and size. The ideal solution, obtained by the *Reasoning Module* is described in Table 9.1.

Table 9.1: Ideal Solution for User 1

Parameter	**Value**
price	40 euro
preparation time	30 minutes
nutrition component importance	1
price component importance	2
time component importance	3
reliability component importance	4
taste component importance	1
aspect component importance	2
smell component importance	3
size component importance	4
energy	1894.86 kcal
carbohydrates	260 g
proteins	106.58 g
fats	130.27 g
vitamin A	1950 ug
vitamin B6	50.65 mg
vitamin C	1045 mg
vitamin D	15 ug
calcium	1600 mg
iron	26.5 mg
sodium	1500 mg

An example of a recommendation that was obtained after running the GSO algorithm is presented in Table 9.2.

Table 9.2: Example of Recommendation for User 1

Meal Type	**Name**
BREAKFAST	Artichoke and Asparagus Salad with Strawberry Dressing
BRUNCH	Amish Sugar Cookies
LUNCH	Armenian Lentil Soup and Arroz De Sabato
SNACK	Amish Sugar Cookies
DINNER	Aloha Chicken with Rice Pilaf with Mushrooms

A series of experiments were conducted on the four algorithms: PSO, PSOS, GSO and GSOS in order to determine which of them gives the best fitness value and the best execution time. There were two cases that were tested: the case in which the number of iterations is small (e.g. 10) and the case in which the number of iterations is large (e.g. 50). For each of the two cases, two graphics are displayed, one for the fitness values and one for the execution time. The number of individuals that compose the swarm population (e.g. the number of particles in the case of PSO and the number of glow-worms in the case of GSO) takes values from the set $\{10, 20, 30, 40, 50\}$.

A comparison of the fitness values obtained for User 1 by running each of the four algorithms with a number of iterations equal with 10 is described in Figure 9.3. The best fitness values are given by the GSO algorithm.

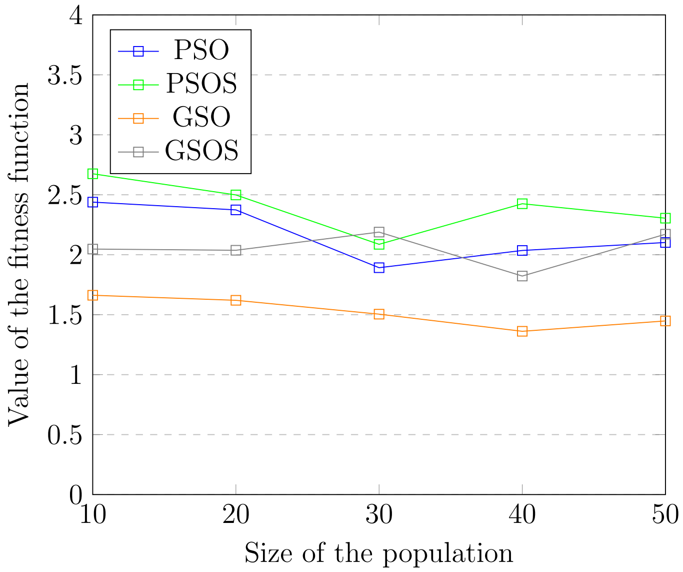

Figure 9.3: Average fitness value of the solution obtained for User 1 by fixing the number of iterations to 10 for every algorithm and by varying the size of the population

The average execution time (seconds) of the algorithms executed for User 1, when the number of iterations is fixed to 10 and the size of the population varies is presented in Figure 9.4. In this case, the classical versions of the algorithms have better execution time values than the versions that are combined with Spark methods.

Another comparison of the fitness values obtained for User 1 by running each of the four algorithms is described in Figure 9.5. In this case the number of iterations is equal with 50 for all the algorithms. The GSO algorithm gives the overall best fitness values.

The average execution time (seconds) of the algorithms executed for User 1, when the number of iterations is fixed to 50 and the size of the population varies is presented in Figure 9.6. In this case, the classical versions of the algorithms have worse execution time values than the versions that are combined with Spark methods.

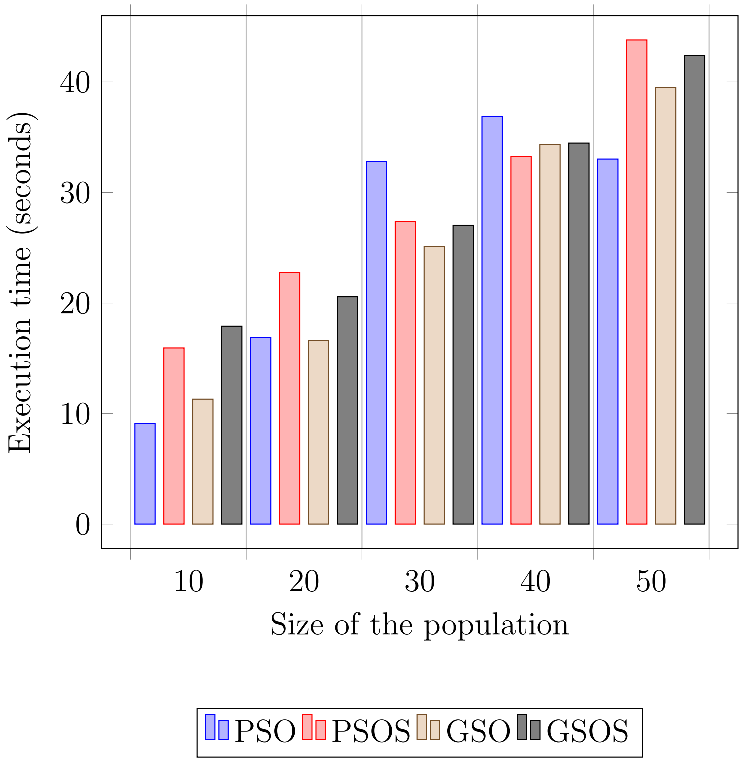

Figure 9.4: Average execution time of the algorithms for User 1 by fixing the number of iterations to 10 for every algorithm and by varying the size of the population

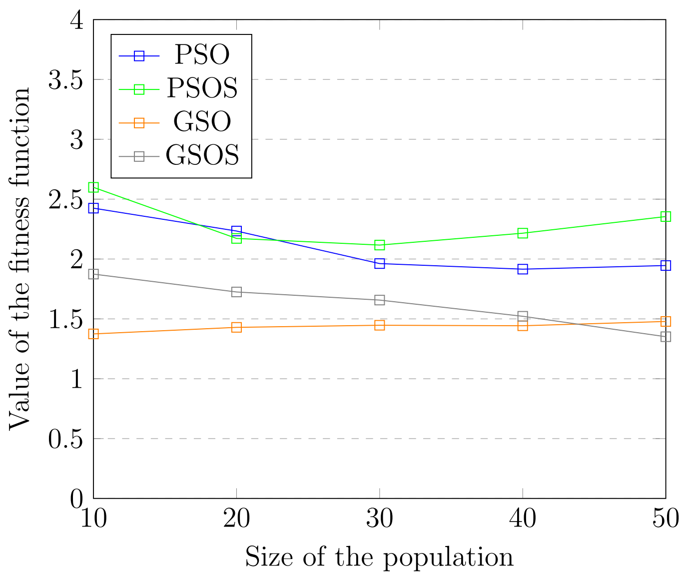

Figure 9.5: Average fitness value of the solution obtained for User 1 by fixing the number of iterations to 50 for every algorithm and by varying the size of the population

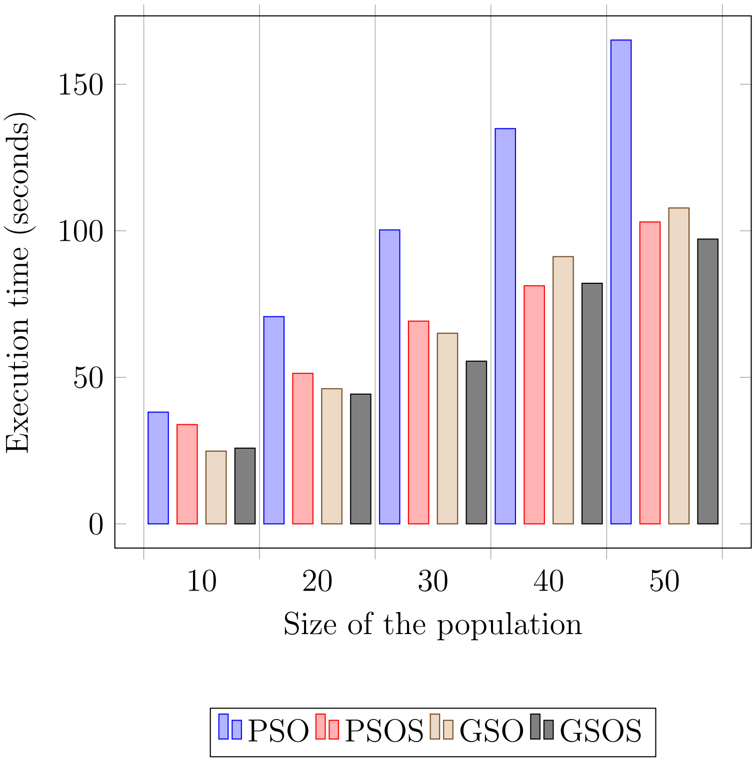

Figure 9.6: Average execution time of the algorithms for User 1 by fixing the number of iterations to 50 for every algorithm and by varying the size of the population

Chapter 10

Conclusions

This thesis presented several optimization techniques for elders malnutrition prevention. The first optimization technique is represented by the generation of the daily nutrient recommendations and diagnostics for an elder using ontologies and reasoning rules, the second optimization technique is represented by the use of two bio-inspired algorithms (PSO and GSO) for the determination of a combination of five meals that match with the profile of the elder and the third optimization technique is represented by the use of techniques from big-data processing to improve the execution time of the proposed bio-inspired algorithms resulting in two modified versions (PSOS and GSOS). The conclusions for each optimization technique are described in Table 10.1.

Table 10.1: Optimization Techniques Conclusions

Optimization Technique	**Conclusions**
Ontologies and reasoning rules	Among the four approaches described in the paper: KAON2, Jena and D2RQ, Jena and R2RML and OWL API and SQL the best performance is given by KAON2. However KAON2 has some limitations in representing the data, and these limitations are overcome by the other approaches.
PSO and GSO	The two proposed algorithms (PSO and GSO) give a recommendation of five meals that matches the profile of an elder in real time. However, the solution is not the ideal solution, but it is close to the ideal one. The experiments conducted on these two approaches showed that GSO performed better on this problem.

PSOS and GSOS	By using techniques from big-data processing, the execution time of PSO and GSO can be reduced. The improvements can be seen when the number of iterations is very large which corresponds to a large number of accesses to the database. This drawback is overcome by the retrieval of the search space only once at the beginning of the algorithm.

As future work, a direction is to adapt the proposed algorithms for the execution on more nodes in a distributed system with different input data sources such as relational databases, .xml files, .json files and ontologies. Another direction is to search for steps of the proposed algorithms that can be executed in parallel. In this thesis the emphasis was more on the retrieval of data and the data structures used to store the information. Finally, another direction is to reduce the overhead of communication between the relational databases and the ontologies in order to use the reasoning engine provided by the ontologies at maximum efficiency.

Bibliography

[1] B. Motik and U. Sattler, "A comparison of reasoning techniques for querying large description logic aboxes," 2006.

[2] M. G. Skjaeveland, "Tutorial on semantic technology at semantic days 2010," *D2R*, 2010.

[3] N. Konstantinou, D. Kouis, and N. Mitrou, "Incremental export of relational database contents into rdf graphs," *International Journal of Database Theory and Application*, no. 33, 2014.

[4] J. Hoddinot, M. Rosegrant, and M. Torero, "Hunger and malnutrition." 2012 Global Copenhagen Consensus, 4 2012, pp. 1–69.

[5] M. Hickson, "Malnutrition and ageing," *Postgrad Med J*, vol. 82, no. 963, pp. 2–8, 1 2006.

[6] L. Donini, P. Scardella, L. Piombo, B. Neri, R. Asprino, A. Proietti, S. Carcaterra, E. Cava, S. Cataldi, D. Cucinotta, G. D. Bella, M. Barbagallo, and A. Morrone, "Malnutrition in elderly: Social and economic determinants," *The Journal of Nutrition, Health & Aging*, vol. 17, no. 1, pp. 9–15, 2013.

[7] C. Evans, "Malnutrition in the elderly: A multifactorial failure to thrive," *The Permanente Journal*, vol. 9, no. 3, pp. 38–41, 2005.

[8] "Today's research on aging, underweight, undernutrition, and the aging," *Program and Policy Implications*, no. 8, pp. 1–5, 2007.

[9] N. Azad, "Nutrition in the elderly," *The Canadian Journal of Diagnosis*, pp. 83–93, 2002.

[10] C. Chia-Hui, L. S. Schilling, and C. H. Lyder, "A concept analysis of malnutrition in the elderly," *Journal of Advanced Nursing*, vol. 36, no. 1, pp. 131–142, 2001.

[11] A.-G. Mamhidir, M. Kihlgren, and V. Soerlie, "Malnutrition in elder care: qualitative analysis of ethical perceptions of politicians and civil servants," *BMC Medical Ethics*, 2010.

[12] K. M. Mogensen and R. A. DiMaria-Ghalili, “Malnutrition in older adults,” vol. 17, no. 9, p. 56, 2015.

[13] J. L. Wells and A. C. Dumbrell, “Nutrition and aging: Assessment and treatment of compromised nutritional status in frail elderly patients,” vol. 1, no. 1, pp. 67–79, 2006.

[14] C.-S. Edgar-Armando, Z.-F. Roberto, T.-F. Yuliana, and P.-N. Alexis, “Diet generator using genetic algorithms,” *Research in Computer Science*, no. 75, pp. 71–77, 2014.

[15] B. K. Seljak, “Dietary menu planning using an evolutionary method,” *Elektrotehniski vestnik*, no. 74, pp. 285–290, 2007.

[16] S. Hartati and S. ’Uyun, “Computation of diet composition for patients suffering from kidney and urinary tract diseases with the fuzzy genetic system,” *International Journal of Computer Applications*, vol. 36, no. 6, pp. 38–45, 2011.

[17] T. Pikes and R. Adams, “Computational nutrition: An algorithm to generate a diet plan to meet specific nutritional requirements,” *E-Health Telecommunication Systems and Networks*, vol. 5, no. 2, pp. 31–38, 2016.

[18] Y. Lv, “Multi-objective nuritional diet optimization based on quantum genetic algorithm,” *2009 Fifth International Conference on Natural Computation*, vol. 4, pp. 336–340, 2009.

[19] S. Gumustenik, T. Senel, and M. A. Cengiz, “A comparative study on bayesian optimization algorithm for nutrition problem,” *Journal of Food and Nutrition Research*, vol. 2, no. 12, pp. 952–958, 2014.

[20] R. Georgieva, “Ontology-based information representation,” 2005.

[21] F. Giunchiglia, I. Zaihrayeu, and F. Farazi, “Converting classifications into owl ontologies,” 2008.

[22] J. V. Fonou-Dombeu and M. Huisman, “Combining ontology development methodologies and semantic web platforms for e-government domain ontology development,” *International Journal of Web & Semantic Technology (IJWesT)*, vol. 2, no. 2, pp. 12–25, 2011.

[23] M. Rockl, K. Frank, P. G. Hermann, and M. T. M. Vera, “Knowledge representation and inference in context-aware computing environments,” *The Second International Conference on Mobile Ubiquitous Computing, Systems, Services, and Technologies*, pp. 89–95, 2008.

[24] B. Chandrasekaran, J. R. Josephson, and V. R. Benjamins, "What are ontologies, and why do we need them," *IEEE Intelligent Systems*, vol. 14, no. 1, pp. 20–26, 1999.

[25] V. Jain and M. Singh, "Ontology development and query retrieval using protege tool," *I. J. Intelligent Systems and Applications*, no. 9, pp. 67–75, 2013.

[26] M. Bergman, "The fundamental importance of keeping an abox and tbox split," 2009.

[27] N. Guarino, D. Oberle, and S. Staab, "What is an ontology," 2009.

[28] G. Karakatsiotis, D. Galanis, G. Lampouras, and I. Androutsopoulos, "Naturalowl: Generating texts from owl ontologies in protege and in second life," 2008.

[29] A. Freitas, D. Schmidt, F. Meneguzzi, R. Vieira, and R. H. Bordini, "Using ontologies as semantic representations of hierarchical task network planning domains," 2014.

[30] A. Ruiz-Iniesta and O. Chorco, "A review of ontologies for describing scholarly and scientific documents," *11th ESWC 2014*, 2014.

[31] A. Doan, J. Madhavan, P. Domingos, and A. Halevy, "Learning to map between ontologies on the semantic web," 2002.

[32] Y. Kalfoglou and M. Schorlemmer, "If-map: An ontology mapping method based on information-flow theory," 2003.

[33] N. Lombard, A. Gerber, and A. van der Merve, "Short paper: Using formal ontologies in the development of countermeasures for military aircraft."

[34] R. Tairas, M. Mernik, and J. Gray, "Using ontologies in the domain analysis of domain-specific languages," *Models in Software Engineering*, pp. 332–342, 2009.

[35] M. Kolchin and D. Zamula, "Food product ontology: Initial implementation of a vocabulary for describing food products," *Proceeding of the 14th Conference of Fruct Association.*

[36] J. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson, "Jena: Implementing the semantic web recommendations," 2003.

[37] D. Jeong, H. Shin, D.-K. Baik, and Y.-S. Jeong, "An efficient web ontology storage considering hierarchical knowledge for jena-based applications," 2009.

[38] K. Jani and D. V. M. Chavda, "A study on semantic web framework: Jena and protege," 2014.

[39] K. Wilkinson, C. Sayers, H. Kuno, and D. Reynolds, "Efficient rdf storage and retrieval in jena2," 2003.

[40] L. Zhong, M. Zheng, J. Yuan, and J. Jin, "The jena-based ontology model inference and retrieval application," *Intelligent Information Management*, pp. 157–160, 2012.

[41] B. McBride, "Jena: Implementing the rdf model and syntax specification," 2001.

[42] H. Rajagopal, "Jena: A java api for ontology management," 2005.

[43] S. Bechhofer, R. Volz, and P. Lord, "Cooking the semantic web with the owl api," *The Semantic Web - ISWC 2003*, pp. 659–675, 2003.

[44] M. Horridge and S. Bechhofer, "The owl api: A java api for owl ontologies," *The Semantic Web - ISWC 2003*, vol. 2, pp. 11–21, 2011.

[45] S. Bechhofer and N. Matentzoglu, "The owl api: An introduction," 2014.

[46] J. Perez, M. Arenas, and C. Gutierrez, "Semantics and complexity of sparql," vol. 34, 2009.

[47] A. Ahmeti and A. Polleres, "Sparql update under rdfs entailment in fully materialized and redundancy-free triples store," 2013.

[48] N. Bikakis, C. Tsinaraki, I. Stavrakantonakis, and S. Christodoulakis, "Supporting sparql update queries in rdf-xml integration," 2014.

[49] T. Eiter, G. Ianni, A. Polleres, R. Schindlauer, and H. Tompits, "Reasoning with rules and ontologies," 2006.

[50] J. Bock, P. Haase, Q. Ji, and R. Volz, "Benchmarking owl reasoners," 2008.

[51] A. Ameen, K. U. R. Khan, and B. P. Rani, "Reasoning in semantic web using jena," vol. 5, no. 4, pp. 39–48, 2014.

[52] S. Bouaicha and Z. Boufaida, "Reasoning on hybrid ontology using extended swrl rules," *JOURNAL OF EMERGING TECHNOLOGIES IN WEB INTELLIGENCE*, vol. 6, no. 1, pp. 3–8, 2014.

[53] A. Ameen, K. U. R. Khan, and B. P. Rani, "Reasoning in semantic web using jena," *Computer Engineering and Intelligent Systems*, vol. 5, no. 4, pp. 39–48, 2014.

[54] X. H. Wang, D. Q. Zhang, T. Gu, and H. K. Pu, "Ontology based context modeling and reasoning using owl," *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshop*, pp. 18–22, 2004.

[55] E. W. Patton and D. L. McGuinness, "A power consumption benchmark for reasoners on mobile devices," *The Semantic Web - ISWC 2014*, pp. 409–424, 2014.

[56] C. Bizer and A. Seaborne, "D2rq - treating non-rdf databases as virtual rdf graphs," 2004.

[57] R. Cyganiak, C. Bizer, J. Garbers, O. Maresch, and C. Becker, "The d2rq mapping language," 2012.

[58] I. Astrova, N. Korda, and A. Kalja, "Rule-based transformation of sql relational databases to owl ontologies," 2007.

[59] Y. Chen and Q. Yang, "Using sparql/update to extend rdb-to-rdf: A mapping approach," *International Journal of Database Theory and Application*, vol. 7, no. 5, pp. 227–238, 2014.

[60] D.-E. Spanos, P. Stavrou, and N. Mitrou, "Bringing relational databases into the semantic web: A survey," *Semantic Web*, vol. 3, pp. 169–209, 2012.

[61] G. Bumans, "Mapping between relational databases and owl ontologies: an example," *SCIENTIFIC PAPERS, UNIVERSITY OF LATVIA*, vol. 756, pp. 99–117, 2010.

[62] M. Rodriguez-Mancha, H. G. Ceballos, F. J. Cantu, and A. Diaz-Prado, "Mapping relational databases through ontology mapping: A case study on information migration," *Proceedings of the 6th International Conference on Ontology Matching*, vol. 814, pp. 244–245, 2011.

[63] W. Hu and Y. Qu, "Discovering simple mappings between relational database schemas and ontologies," *The Semantic Web*, vol. 4825, pp. 225–238, 2007.

[64] A. Gali, C. X. Chen, K. T. Claypool, and R. Uceda-Sosa, "From ontology to relational databases," *Conceptual Modeling for Advanced Application Domains*, vol. 3289, pp. 278–289, 2004.

[65] F. Michel, J. Montagnat, and C. Faron-Zucker, "A survey of rdb to rdf translation approaches and tools," 2014.

[66] I. Astrova and B. Stantic, "Reverse engineering of relational databases to ontologies: An approach based on an analysis of html forms."

[67] M. Seleng, M. Laclavik, Z. Balogh, and L. Hluchy, "Rdb2onto: Approach for creating semantic metadata from relational database data," 2015.

[68] I. F. Jr., X.-S. Yang, I. Fister, J. Brest, and D. Fister, "A brief review of nature-inspired algorithms for optimization," *Elektrotehniski Vestnik*, vol. 80, no. 3, pp. 1–7, 2013.

[69] J. Bongard, "Biologically inspired computing," 2009.

[70] M. Breza and J. A. McCann, "Lessons in implementing bio-inspired algorithms on wireless sensor networks," *Conference on Adaptive Hardware and Systems*, 2008.

[71] R. V. Kulkarni, G. K. Venayagamoorthy, and M. X. Cheng, "Bio-inspired node localization in wireless sensor networks," *Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics*, 2009.

[72] L. Wang, J. Shen, and J. Yong, "A survey on bio-inspired algorithms for web service composition," *Proceeding of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design*, 2012.

[73] B. S. and S. S. Sathya, "A survey of bio inspired optimization algorithms," *International Journal of Soft Computing and Engineering (IJSCE)*, vol. 2, no. 2, pp. 137–151, 2012.

[74] F. Dressler, B. Kruger, G. Fuchs, and R. German, "Self-organization in sensor networks using bio-inspired mechanisms," *Proceeding of the 3rd International Conference on Bio-Inspired Models of Network, Information and Computing Systems*, no. 10, 2008.

[75] R. V. Kulkarni and G. K. Venayagamoorthy, "Bio-inspired algorithms for autonomous deployment and localization of sensor nodes," *IEEE Transactions on Systems, Man and Cybernetics-Part C: Applications and Reviews*, vol. 40, no. 6, 2010.

[76] T. Renk, C. Kloeck, D. Burgkhardt, F. K. Jondral, D. Grandblaise, S. Gault, and J. C. Dunat, "Bio-inspired algorithms for dynamic resource allocation in cognitive wireless networks," *Mobile Netw Appl*, 2008.

[77] E. Mezura-Montes and B. C. Lopez-Ramirez, "Comparing bio-inspired algorithms in constrained optimization problems," 2007.

[78] X. Jun and H. Chang, "The discrete binary version of the improved particle swarm optimization algorithm," 2009.

[79] Y. LUO and X. CHE, "Chaos immune particle swarm optimization algorithm with hybrid discrete variables and its application to mechanical optimization," *Third International Symposium on Intelligent Information Technology Application Workshop*, 2009.

[80] X. Hu, L. Wang, and Y. Zhong, "An improved particle swarm optimization algorithm for site index curve model," 2011.

[81] Y. Fu, M. Ding, and C. Zhou, "Phase angle-encoded and quantum-behaved particle swarm optimization applied to three-dimensional route planning for uav," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 42, no. 2, pp. 511–526, 2012.

[82] C. Ji, F. Liu, and X. Zhang, "Particle swarm optimization based on catfish effect for flood optimal operation of reservoir," *Seventh International Conference on Natural Computation*, pp. 1197–1201, 2011.

[83] X. Wu, "A density adjustment based particle swarm optimization learning algorithm for neural network design," pp. 2829–2832, 2011.

[84] I. Ibrahim, Z. M. Yusof, S. W. Nawawi, M. A. A. Rahim, K. Khalil, H. Ahmad, and Z. Ibrahim, "A novel multi-state particle swarm optimization for discrete combinatorial optimization problems," *Fourth International Conference on Computational Intelligence, Modelling and Simulation*, pp. 18–23, 2012.

[85] Y. Zhe-ping, D. Chao, Z. Jia-jia, and C. Dong-nan, "A novel two-subpopulation particle swarm optimization," *Proceedings of the 10th World Congress on Intelligent Control and Automation*, pp. 4113–4117, 2012.

[86] Z. Li and T. Zhu, "Research on global-local optimal information ratio particle swarm optimization for vehicle scheduling problem," *7th International Conference on Intelligent Human-Machine Systems and Cybernetics*, pp. 92–96, 2015.

[87] J. Maltese, B. M. Ombuki-Berman, and A. P. Engelbrecht, "Co-operative vector evaluated particle swarm optimization for multi-objective optimization," *IEEE Symposium Series on Computational Intelligence*, pp. 1294–1301, 2015.

[88] Y. Zeng and J. Zhang, "Glowworm swarm optimization and heuristic algorithm for rectangle packing problem," *2012 IEEE International Conference on Information Science and Technology, Wuhan, Hubei, China*, vol. 2, no. 12, pp. 136–140, 2012.

[89] N. Al-Madi, I. Aljarah, and S. A. Ludwig, "Parallel glowworm swarm optimization clustering algorithm based on mapreduce," 2014.

[90] J. Gu and K. Wen, "Glowworm swarm optimization algorithm with quantum-behaved properties," *2014 10th International Conference on Natural Computation*, pp. 430–436, 2014.

[91] H. Deng-xu, L. Gui-qing, and Z. Hua-zheng, "Glowworm swarm optimization algorithm for solving multi-objective optimization problem," *2013 Ninth International Conference on Computational Intelligence and Security*, pp. 11–15, 2013.

[92] X. Lu and W. Sun, "An improved self-adapting glowworm swarm optimization algorithm," 2013.

[93] L. He, X. Tong, and S. Huang, "A glowworm swarm optimization algorithm with improved movement rule," *2012 Fifth International Conference on Intelligent Networks and Intelligent Systems*, 2012.

[94] L. He, X. Tong, S. Huang, and Q. Wang, "Glowworm swarm optimization algorithm with improved movement pattern," *2013 6th International Conference on Intelligent Networks and Intelligent Systems*, pp. 43–46, 2013.

[95] L. Jing, H. Song, and X. Lv, "Research and application on job shop planning based on improved glowworm swarm optimization algorithm," *2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics*, pp. 139–143, 2013.

[96] W. Zheng, H. Yin, J. Fu, P. Fu, B. Liu, and W. Pan, "Range expansion of mobile wireless system by cooperative transmission based on glowworm swarm optimization," 2014.

[97] X. Meng and A. Talwalkar, "Mllib: Machine learning in apache spark," *Journal of Machine Learning Research*, pp. 1–7, 2016.

[98] T. Chiba and T. Onodera, "Workload characterization and optimization of tpc-h queries on apache spark," 2016.

[99] G. Zhao and C. L. D. Sun, "Sparksw: scalable distributed computing system for large-scale biological sequence alignment," *15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 845–852, 2015.

[100] N. Bharill, A. Tiwari, and A. Malviya, "Fuzzy based clustering algorithms to handle big data with implementation on apache spark," *2016 IEEE Second International Conference on Big Data Computing Services and Applications*, pp. 95–104, 2016.

[101] M. A. Alsheikh, D. Niyato, S. Lin, H.-P. Tan, and Z. Han, "Mobile big data analytics using deep learning and apache spark," 2016.

[102] M. Solaimani, R. Gopalan, L. Khan, P. T. Brandt, and B. Thuraisingham, "Spark-based political event coding," *2016 IEEE Second International Conference on Big Data Computing Services and Applications*, pp. 14–23, 2016.

[103] X. Lu, M. W. ur Rahman, N. Islam, D. Shankar, and D. K. D. Panda, "Accelerating spark with rdma for big data processing: Early experiences," *2014 IEEE 22nd Annual Symposium on High-Performance Interconnects*, pp. 9–16, 2014.

[104] W. Huang, L. Meng, D. Zhang, and W. Zhang, "In-memory parallel processing of massive remotely sensed data using an apache spark on hadoop yarn model," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, pp. 1–17, 2016.

[105] Y. Yan, L. Huang, and L. Yi, "Is apache spark scalable to seismic data analytics and computations?" *2015 IEEE International Conference on Big Data (Big Data)*, pp. 2036–2045, 2015.

[106] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," 2010.

[107] M. Solaimani, M. Iftekhar, L. Khan, and B. Thuraisingham, "Statistical technique for online anomaly detection using spark over heterogeneous data from multi-source vmware performance data," *2014 IEEE International Conference on Big Data*, pp. 1086–1094, 2014.

[108] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," 2012.

[109] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," *COMMUNICATIONS OF THE ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[110] H. Karau, A. Knowinski, P. Wendell, and M. Zaharia, "Learning spark lightning-fast data analysis," 2015.

[111] H. Mushtaq and Z. Al-Ars, "Cluster-based apache spark implementation of the gatk dna analysis pipeline," *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 1471–1477, 2015.

[112] M. Winlaw, M. B. Hynes, A. Caterini, and H. D. Sterck, "Algorithmic acceleration of parallel als for collaborative filtering: Speeding up distributed big data recommendation in spark," *2015 IEEE 21st International Conference on Parallel and Distributed Systems*, pp. 682–691, 2015.

[113] M. Zaharia, D. Borthakur, J. S. Sarma, K. Elmeleegy, S. Shenker, and I. Stoica, "Delay scheduling: A simple technique for achieving locality and fairness in cluster scheduling," *Proceedings of the 5th European Conference on Computer systems*, pp. 265–278, 2010.

[114] D. Harnie, A. E. Vapirev, J. K. Wegnet, A. Gedich, M. Steijaert, R. Wuyts, and W. D. Meuter, "Scaling machine learning for target prediction in drug discovery using apache spark," *2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 871–879, 2015.

[115] A. I. Maarala, M. Rautiainen, M. Salmi, S. Pirttikangas, and J. Riekki, "Low latency analytics for streaming traffic data with apache spark," *2015 IEEE International Conference on Big Data (Big Data)*, pp. 2855–2858, 2015.

[116] J. Shi, Y. Qiu, U. F. Minhas, L. Jiao, C. Wang, B. Reinwald, and F. Ozcan, "Clash of the titans: Mapreduce vs. spark for large scale data analytics," pp. 2110–2121, 2015.

[117] "Diet4elders aal project," www.diet4elders.eu.

[118] "Mccance and widdowson's the composition of foods: Seventh summary edition," http://pubs.rsc.org/en/content/ebook/9781849736367#!divbookcontent.

[119] "Physical activity level," http://www.nhlbi.nih.gov/health/educational/lose_wt/BMI/bmicalc.htm/.

[120] D. Moldovan, M. Antal, D. Valea, C. Pop, T. Cioara, I. Anghel, and I. Salomie, "Tools for mapping ontologies to relational databases: A comparative evaluation," *2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, pp. 77–83, 2015.

[121] R.-M. Bonta, "Food ordering system for older adults based on bio-inspired techniques techniques for generating menu recommendations using the weed optimization and particle swarm optimization algorithms."

[122] K. N. Krishnanand and D. Ghose, "Glowworm swarm optimization for searching higher dimensional spaces," *Innovations in Swarm Intelligence*, pp. 61–75, 2009.

Appendix A

Acronyms

ACO	Ant Colony Optimization
API	Application Programming Interface
BFA	Bacterial Foraging Algorithm
BMI	Body Mass Index
BOA	Bayesian Optimization Algorithm
CEPSO	Catfish Effect Particle Swarm Optimization
CIPSO	Chaos Immune Particle Swarm Optimization
CPU	Central Processing Unit
DAG	Directed Acyclic Graph
DIG	DL Implementors Group
DL	Description Logic
DNA	Deoxyribonucleic Acid
EA	Evolutionary Algorithm
GA	Genetic Algorithm
GB	Gigabyte
GHZ	Gigahertz
GLIRPSO	Global-local Optimal Information Ratio Particle Swarm Optimization
GSO	Glow-worm Swarm Optimization
GSOS	Glow-worm Swarm Optimization using Spark
HDFS	Hadoop Distributed File System
HTML	Hypertext Markup Language
IIS	Iterative Improvement Strategy
IMGSO	Improved Movement Patter Glow-worm Swarm Optimization
JVM	Java Virtual Machine
KTS	Knowledge Transfer Strategy
MI	Maastricht Index
MOP	Multi-objective Optimization Problem
MRCGSO	MapReduce Clustering Glow-worm Swarm Optimization

MSPSO	Multi-state Particle Swarm Optimization
NLG	Natural Language Generation
NP	Nondeterministic Polynomial Time
NRI	Nutritional Risk Index
OS	Operating System
OWL	Web Ontology Language
PAF	Physical Activity Factor
PSO	Particle Swarm Optimization
PSOS	Particle Swarm Optimization using Spark
RAM	Random Access Memory
RDB	Relational Database
RDBMS	Relational Database Management System
RDD	Resilient Distributed Dataset
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
SPARQL	Simple Protocol and RDF Query Language
SQL	Structure Query Language
SW	Smith-Waterman
SWRL	Semantic Web Rule Language
TSPSO	Two Subpopulation Particle Swarm Optimization
QGA	Quantum Genetic Algorithm
QGSO	Quantum-behaved Glow-worm Swarm Optimization
VEPSO	Vector-Evaluated Particle Swarm Optimization
WWW	World Wide Web Consortium
WWW	World Wide Web
XML	Extensible Markup Language

Appendix B

Relevant Code

Listing B.1: Jena Rules

```
@prefix : <http://www.semanticweb.org/ontologies/diagnostic#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

[NormalWeightRule:
  (?x rdf:type :User),
  (?y rdf:type :Measurements),
  (?y :hasUser ?x),
  (?y :hasBMI ?b),
  ge(?b, 18.5),
  lessThan(?b, 25.0),
->
  (?x :hasDiagnostic :NormalWeight),
]

[BMIRule:
  (?x rdf:type :Measurements),
  (?x :hasHeight ?h),
  (?x :hasWeight ?w),
  product(?h, ?h, ?n),
  product(?n, 1.0, ?m),
  quotient(?w, ?m, ?r),
  product(?r, 10000.0, ?rez),
->
  (?x :hasBMI ?rez),
]

[ObeseRule:
  (?x rdf:type :User),
  (?y rdf:type :Measurements),
  (?y :hasUser ?x),
  (?y :hasBMI ?b),
  ge(?b, 30.0),
->
  (?x :hasDiagnostic :Obese),
]

[UnderweightRule:
  (?x rdf:type :User),
  (?y rdf:type :Measurements),
  (?y :hasUser ?x),
  (?y :hasBMI ?b),
  lessThan(?b, 18.5),
```

```
->
  (?x :hasDiagnostic :Underweight),
]

[OverweightRule:
  (?x rdf:type :User),
  (?y rdf:type :Measurements),
  (?y :hasUser ?x),
  (?y :hasBMI ?b),
  ge(?b, 25.0),
  lessThan(?b, 30.0),
->
  (?x :hasDiagnostic :Overweight),
]
```

Listing B.2: R2RML Mapping

```
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix ex: <http://example.com/ns#> .
@prefix elders: <http://www.semanticweb.org/ontologies/2015/2/elders#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@base <http://www.semanticweb.org/ontologies/2015/2/elders/> .

<EldersTableView> rr:sqlQuery """
␣␣␣␣␣␣␣SELECT␣CONCAT('User_',␣user_id)␣AS␣userId
␣␣␣␣␣␣␣␣␣␣␣␣,␣user_id
␣␣␣␣␣␣␣␣␣␣␣␣,␣firstName
␣␣␣␣␣␣␣␣␣␣␣␣,␣lastName
␣␣␣␣␣␣␣␣␣␣␣␣,␣gender
␣␣␣␣␣␣␣␣␣␣␣␣,␣age
␣␣␣␣␣␣␣FROM␣elders.USERS;
␣␣␣␣␣␣␣""" .

<TriplesMap1>
    a rr:TriplesMap;
    rr:logicalTable <EldersTableView>;

    rr:subjectMap [ rr:column "userId"; ];

    rr:predicateObjectMap
    [
      rr:predicate              elders:hasFirstName ;
      rr:objectMap      [ rr:column "\"firstName\""; ]
    ];

    rr:predicateObjectMap
    [
      rr:predicate              elders:hasLastName ;
      rr:objectMap      [ rr:column "\"lastName\""; ]
    ];

    rr:predicateObjectMap
    [
      rr:predicate              elders:hasGender ;
      rr:objectMap      [ rr:column "\"gender\""; ]
    ];

    rr:predicateObjectMap
    [
      rr:predicate              elders:hasAge ;
      rr:objectMap      [ rr:column "\"age\""; ]
    ]
```

```
.

<ValuesTableView> rr:sqlQuery """
␣␣␣␣␣␣␣SELECT␣CONCAT('Values',␣id)␣AS␣valuesId
␣␣␣␣␣␣␣␣␣␣␣␣,␣user_id
␣␣␣␣␣␣␣␣␣␣␣␣,␣weight
␣␣␣␣␣␣␣␣␣␣␣␣,␣height
␣␣␣␣␣␣␣FROM␣elders.VALUES;
␣␣␣␣␣␣""" .

<TriplesMap2>
    a rr:TriplesMap;
    rr:logicalTable <ValuesTableView>;

    rr:subjectMap [ rr:column "valuesId"; ];

    rr:predicateObjectMap
    [
      rr:predicate              elders:hasWeight ;
      rr:objectMap     [ rr:column "weight"; ]
    ];

     rr:predicateObjectMap
    [
      rr:predicate              elders:hasHeight ;
      rr:objectMap     [ rr:column "height"; ]
    ];

    rr:predicateObjectMap [
      rr:predicate              elders:hasUser ;
      rr:objectMap     [
        a rr:RefObjectMap ;
        rr:parentTriplesMap <TriplesMap1>;
        rr:joinCondition [
          rr:child "user_id";
          rr:parent "user_id";
        ];
      ];
    ]
    .
```

Listing B.3: D2RQ Mapping

```
# D2RQ Namespace
@prefix d2rq: <http://www.wiwiss.fu-berlin.de/suhl/bizer/D2RQ/0.1#> .
# Namespace of the ontology
@prefix elders: <http://www.semanticweb.org/ontologies/2015/2/elders#> .
# Namespace of the mapping file; does not appear in mapped data
@prefix map: <#> .
# Other namespaces
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix jdbc: <http://d2rq.org/terms/jdbc/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

@prefix vocab: <vocab/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix iswc: <http://annotation.semanticweb.org/iswc/iswc.daml#> .
@prefix vcard: <http://www.w3.org/2001/vcard-rdf/3.0#> .
```

```
@prefix jdbc: <http://d2rq.org/terms/jdbc/> .

map:database a d2rq:Database;
    d2rq:jdbcDSN "jdbc:mysql://localhost/elders";
    d2rq:jdbcDriver "com.mysql.jdbc.Driver";
    d2rq:username "root";
    d2rq:password "admin";
    jdbc:autoReconnect "true";
        jdbc:zeroDateTimeBehavior "convertToNull";
        .

##########################################################################
######################### users ##########################################
##########################################################################

# Table users
map:users a d2rq:ClassMap;
        d2rq:dataStorage map:database;
        d2rq:uriPattern "http://www.semanticweb.org/ontologies/2015/2/elders#User_@@users.
            user_id@@";
        d2rq:class elders:User;
        .

# Data Property firstName
map:firstName a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:users;
    d2rq:property elders:hasFirstName;
    d2rq:column "users.firstName";
    d2rq:datatype xsd:string;
    .

# Data Property lastName
map:lastName a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:users;
    d2rq:property elders:hasLastName;
    d2rq:column "users.lastName";
    d2rq:datatype xsd:string;
    .

# Data Property gender
map:gender a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:users;
    d2rq:property elders:hasGender;
    d2rq:column "users.gender";
    d2rq:datatype xsd:string;
        .

# Data Property age
map:age a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:users;
    d2rq:property elders:hasAge;
    d2rq:column "users.age";
    d2rq:datatype xsd:int;
    .

##########################################################################
######################### values #########################################
##########################################################################

# Table measurements
map:values a d2rq:ClassMap;
        d2rq:dataStorage map:database;
        d2rq:uriPattern "http://www.semanticweb.org/ontologies/2015/2/elders#
```

```
            Values@@values.id@@";
        d2rq:class elders:Values;
        .

# Object property hasUser
map:hasUser a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:values;
    d2rq:property elders:hasUser;
    d2rq:refersToClassMap map:users;
    d2rq:join "values.id␣=␣values_2.id";
        d2rq:join "values.user_id␣=␣users.user_id";
        d2rq:alias "values␣as␣values_2";
    .

# Data Property weight
map:weight a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:values;
    d2rq:property elders:hasWeight;
    d2rq:column "values.weight";
    d2rq:datatype xsd:int;
        .

# Data Property height
map:height a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:values;
    d2rq:property elders:hasHeight;
    d2rq:column "values.height";
    d2rq:datatype xsd:int;
        .
```

Listing B.4: KAON2 Mapping

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<db:DBOntology db:name="http://www.semanticweb.org/ontologies/2015/2/elders" xmlns:db="
    http://kaon2.semanticweb.org/db#">

<db:Database db:connectionString="jdbc:mysql://localhost/elders" db:userName="root"
    db:password="admin" db:driverClassName="com.mysql.jdbc.Driver"/>

<db:OWLClass db:name="http://www.semanticweb.org/ontologies/2015/2/elders#User">
    <db:Table db:tableName="users">
        <db:String db:fieldName="user_id" db:uriPrefix="http://www.semanticweb.org/
            ontologies/2015/2/elders#User_" db:primaryKey="true"/>
    </db:Table>
</db:OWLClass>

<db:DatatypeProperty db:name="http://www.semanticweb.org/ontologies/2015/2/elders#
    hasFirstName">
    <db:Table db:tableName="users">
        <db:String db:fieldName="user_id" db:uriPrefix="http://www.semanticweb.org/
            ontologies/2015/2/elders#User_" db:primaryKey="true"/>
        <db:String db:fieldName="firstName"/>
    </db:Table>
</db:DatatypeProperty>

<db:DatatypeProperty db:name="http://www.semanticweb.org/ontologies/2015/2/elders#
    hasLastName">
    <db:Table db:tableName="users">
        <db:String db:fieldName="user_id" db:uriPrefix="http://www.semanticweb.org/
            ontologies/2015/2/elders#User_" db:primaryKey="true"/>
        <db:String db:fieldName="lastName"/>
    </db:Table>
```

```
</db:DatatypeProperty>

<db:DatatypeProperty db:name="http://www.semanticweb.org/ontologies/2015/2/elders#hasGender">
    <db:Table db:tableName="users">
        <db:String db:fieldName="user_id" db:uriPrefix="http://www.semanticweb.org/ontologies/2015/2/elders#User_" db:primaryKey="true"/>
        <db:String db:fieldName="gender"/>
    </db:Table>
</db:DatatypeProperty>

<db:DatatypeProperty db:name="http://www.semanticweb.org/ontologies/2015/2/elders#hasAge">
    <db:Table db:tableName="users">
        <db:String db:fieldName="user_id" db:uriPrefix="http://www.semanticweb.org/ontologies/2015/2/elders#User_" db:primaryKey="true"/>
        <db:Integer db:fieldName="age"/>
    </db:Table>
</db:DatatypeProperty>

<db:OWLClass db:name="http://www.semanticweb.org/ontologies/2015/2/elders#Values">
    <db:Table db:tableName="elders.values">
        <db:IndividualInteger db:fieldName="id" db:uriPrefix="http://www.semanticweb.org/ontologies/2015/2/elders#Values" db:primaryKey="true"/>
    </db:Table>
</db:OWLClass>

<db:DatatypeProperty db:name="http://www.semanticweb.org/ontologies/2015/2/elders#hasWeight">
    <db:Table db:tableName="elders.values">
        <db:IndividualInteger db:fieldName="id" db:uriPrefix="http://www.semanticweb.org/ontologies/2015/2/elders#Values" db:primaryKey="true"/>
        <db:Double db:fieldName="weight"/>
    </db:Table>
</db:DatatypeProperty>

<db:DatatypeProperty db:name="http://www.semanticweb.org/ontologies/2015/2/elders#hasHeight">
    <db:Table db:tableName="elders.values">
        <db:IndividualInteger db:fieldName="id" db:uriPrefix="http://www.semanticweb.org/ontologies/2015/2/elders#Values" db:primaryKey="true"/>
        <db:Double db:fieldName="height"/>
    </db:Table>
</db:DatatypeProperty>

<db:ObjectProperty db:name="http://www.semanticweb.org/ontologies/2015/2/elders#hasUser">
    <db:Table db:tableName="elders.values">
        <db:IndividualInteger db:fieldName="id" db:uriPrefix="http://www.semanticweb.org/ontologies/2015/2/elders#Values" db:primaryKey="true"/>
        <db:String db:fieldName="user_id" db:uriPrefix="http://www.semanticweb.org/ontologies/2015/2/elders#User_" db:primaryKey="true"/>
    </db:Table>
</db:ObjectProperty>

</db:DBOntology>
```

Listing B.5: SPARQL Query

```
String sparqlForBasicFood = DIET4EldersOntology.PREFIX
                + "SELECT␣?foodIntake␣(SUM(xsd:double(?coefficient)␣*␣xsd:double(?grams))␣AS␣?Grams)␣"
                + "␣WHERE␣{"
                + "␣␣␣␣?foodIntake␣nutritionmonitoring:hasUsername␣\""
                + username
```

```
                    + "\" . "
                    + "    ?foodIntake nutritionmonitoring:hasDate ?date "
                    + "    FILTER(xsd:dateTime(?date) >= \""
                    + startString
                    + "\"^^xsd:dateTime && "
                    + "    xsd:dateTime(?date) <= \""
                    + endString
                    + "\"^^xsd:dateTime) "
                    + "    ?foodQuantityInFoodIntake nutritionassessment:foodQuantityHasFoodIntake ?foodIntake . "
                    + "    ?foodQuantityInFoodIntake nutritionassessment:foodIntakeHasFoodQuantity ?foodQuantity . "
                    + "    ?foodQuantityInFoodIntake nutritionassessment:foodQuantityInFoodIntakeHasCoefficient ?coefficient . "
                    + "    ?foodQuantity nutritionassessment:hasGramsValue ?grams . "
                    + "    ?foodQuantity nutritionassessment:hasBasicFood ?basicFood . "
                    + "    { ?basicFood rdf:type food:Smoked_fish } "
                    + "    UNION "
                    + "    { ?basicFood rdf:type food:Other_fish } "
                    + "    UNION "
                    + "    { ?basicFood rdf:type food:Pickled_fish } "
                    + "    UNION "
                    + "    { ?basicFood rdf:type food:Dried_and_salted_fish } "
                    + "    UNION "
                    + "    { ?basicFood rdf:type food:Fish_products } "
                    + "    UNION "
                    + "    { ?basicFood rdf:type food:Canned_fish } "
                    + "    UNION "
                    + "    { ?basicFood rdf:type food:Restructured_fish_and_fish_analogues } "
                    + " } GROUP BY ?foodIntake ";
```

Listing B.6: Spark code in Java

```
private DataFrame packetEntitiesDataFrame;

private JavaRDD<PacketEntity> breakfastRDD;
private JavaRDD<PacketEntity> lunchRDD;
private JavaRDD<PacketEntity> dinnerRDD;
private JavaRDD<PacketEntity> snackRDD;

private DataFrame relationPacketRecipe;

private DataFrame recipeFoodRelation;

private DataFrame foodDescription;

public void initialize() {
        Map<String, String> options = new HashMap<String, String>();
        options.put("driver", Utils.DRIVER);
        options.put("url", Utils.URL + "?user=" + Utils.USERNAME + "&password=" + Utils.PASSWORD);
        options.put("dbtable",
                        "(SELECT * FROM packet JOIN relation_packet_distributor JOIN relation_distributor_ga WHERE "
                                    + "packet.idPacket = relation_packet_distributor.packetId AND "
                                    + "relation_distributor_ga.distrID = relation_packet_distributor.distributorId) as packet");

        packetEntitiesDataFrame = Utils.sqlContext.read().format("jdbc").options(options).load();

```

```
        packetEntitiesDataFrame.cache();

        breakfastRDD = packetEntitiesDataFrame.javaRDD().filter((new Function<Row, Boolean>() {
            /**
             *
             */
            private static final long serialVersionUID = 1L;

            public Boolean call(Row row) throws Exception {
                if (row.getInt(18) == 0) {
                    return true;
                } else {
                    return false;
                }
            }
        })).map(new PacketEntityFunction()).cache();

        lunchRDD = packetEntitiesDataFrame.javaRDD().filter((new Function<Row, Boolean>() {
            /**
             *
             */
            private static final long serialVersionUID = 1L;

            public Boolean call(Row row) throws Exception {
                if (row.getInt(18) == 1) {
                    return true;
                } else {
                    return false;
                }
            }
        })).map(new PacketEntityFunction()).cache();

        dinnerRDD = packetEntitiesDataFrame.javaRDD().filter((new Function<Row, Boolean>() {
            /**
             *
             */
            private static final long serialVersionUID = 1L;

            public Boolean call(Row row) throws Exception {
                if (row.getInt(18) == 2) {
                    return true;
                } else {
                    return false;
                }
            }
        })).map(new PacketEntityFunction()).cache();

        snackRDD = packetEntitiesDataFrame.javaRDD().filter((new Function<Row, Boolean>() {
            /**
             *
             */
            private static final long serialVersionUID = 1L;

            public Boolean call(Row row) throws Exception {
                if (row.getInt(18) == 2) {
                    return true;
                } else {
                    return false;
                }
```

```
                }
        })).map(new PacketEntityFunction()).cache();

        options.put("dbtable", "relation_packet_recipe");

        relationPacketRecipe = Utils.sqlContext.read().format("jdbc").options(options).
            load();

        relationPacketRecipe.cache();

        options.put("dbtable", "recipe_food_relation");

        recipeFoodRelation = Utils.sqlContext.read().format("jdbc").options(options).load
            ();

        recipeFoodRelation.cache();

        options.put("dbtable", "food_descr");

        foodDescription = Utils.sqlContext.read().format("jdbc").options(options).load();

        foodDescription.cache();

}
```

Appendix C

Published Papers

1 **Dorin Moldovan**, Claudia Pop, Marcel Antal, Tudor Cioara, Ionut Anghel, Ioan Salomie, *"SWAG: Semantic Web Application Generator - A Library for Using Ontologies as Web Services"*, Intelligent Computing Computer and Processing (ICCP) ICCP 2016, Cluj-Napoca, **submitted**

2 Claudia Pop, Alexandra Craciun, Carla Knoblau, Marcel Antal, **Dorin Moldovan**, Tudor Cioara, Ionut Anghel, Ioan Salomie, *"Semantic Data Factory: A Framework for Using Domain Knowledge in Software Application Development"*, Intelligent Computing Computer and Processing (ICCP) ICCP 2016, Cluj-Napoca, **submitted**

3 Viorica Rozina Chifu, Ioan Salomie, Laura Petrisor, Emil Stefan Chifu, **Dorin Moldovan**, *"Hybrid Immune based Method for Generating Healthy Meals for Older Adults"*, SYNASC 2016, Timisoara, **submitted**

4 **Dorin Moldovan**, Marcel Antal, Dan Valea, Claudia Pop, Tudor Cioara, Ionut Anghel, Ioan Salomie, *"Tools for Mapping Ontologies to Relational Databases: A Comparative Evaluation"*, Intelligent Computing Computer and Processing (ICCP) ICCP 2015, Cluj-Napoca

5 Claudia Pop, **Dorin Moldovan**, Marcel Antal, Dan Valea, Tudor Cioara, Ionut Anghel, Ioan Salomie, *"M2O: A Library for Using Ontologies in Software Engineering"*, Intelligent Computing Computer and Processing (ICCP) ICCP 2015, Cluj-Napoca

6 Cristina Bianca Pop, Viorica Rozina Chifu, Ioan Salomie, Cristian Prigoana, Tiberiu Boros, **Dorin Moldovan**, *"Generating Healthy Menus for Older Adults using a Hybrid Honey Bees Mating Optimization Approach"*, SYNASC 2015, Timisoara