DACSR: Dietetic Advice Construction After Stomach Reduction

Deagan Otterspeer and Skip Overgoor
Vrije Universiteit, 1081 HV Amsterdam

January 6, 2019

1 Introduction

In this report, the creation of a knowledge-based system for dietetic advice construction for patients which had a stomach reduction is depicted. Dietitians nowadays construct all the dietetic advice for their patients from scratch. We introduce a system which creates this advice, based on a number of values given by the patient, and external lab values measured in the blood of the patient. This system will help dietitians with most of the repetitive advice construction on a daily base.

This report starts out by presenting the context and organization models, followed by the knowledge model. Consecutively, the reuse opportunities of knowledge through the use of ontologies is described, followed by the inference structure and the task model. Afterwards, two example scenarios are stated. The communication model can be seen behind those examples, followed by the implementation process. Finally, the methods, and a reflection of our journey is given.

2 Context and Organization Models

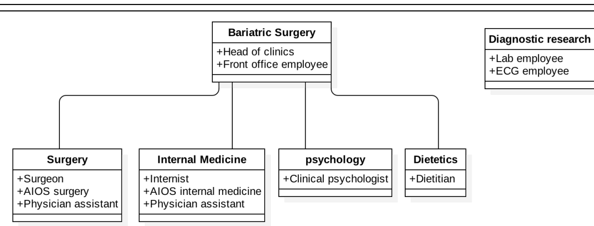
The knowledge-based (KB) system proposed is concerned with the domain of dietetics, as a sub-domain of healthcare. As a finalizing component of stomach reduction surgeries, patients have meetings with a dietitian to both acquire advice on how to treat certain complaints, and to gain insights in the nutritional intake values to guide the recovery and weight-loss process. As such, the system can be used by hospitals with a dietetics department, independent dietetic firms, to aid in this process. The organizational models are depicted below. These models describe the organizations and specific agents involved in the process aided by the system.

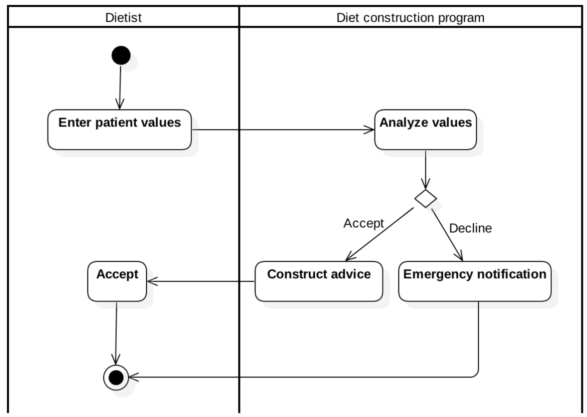
OM-1
Identifying knowledge-oriented problems and opportunities in the organization

Organization Model	**Problems and Opportunities Worksheet OM-1**
PROBLEMS AND OPPORTUNITIES	A lot of dietetic advice is standard, and according to procedure. Due to this fact, a system could help with constructing a specific diet, according to the values of a patient after a stomach reduction. Another problem is that the patients often are checked on in groups, due to lack of time to see each patient individually. This is mostly due to the standard procedures, which are vital.
ORGANIZATIONAL CONTEXT	1. The organization is a hospital. A hospital does not have commercial goals. The main goal is to heal the patient. If the patient gets bad advice from the dietitian, the patient will return to the hospital. This will result in the patient being dissatisfied, and maybe even in another operation 2. The hospital mainly has to work with insurance companies, and other hospitals 3. The strategy of the hospital is to heal the patient as effective and easy as possible. 4. Value is delivered through: proper diets, and complication solutions. Hospital value comes from: image (lowcomplaint/failure rates/expertise), low cost suppliers, time/cost-efficiency(re-use of previous work).
SOLUTIONS	A possible solution could be a system that checks if the patient needs a standard diet, or needs more complicated diet restrictions. An addition to this system could be that certain dietetic values (macronutrients, for example) are predicted.

OM-2

Description of organizational aspects that have an impact on and/or are affected by chosen knowledge solutions

Organization Model	**Variant Aspects Worksheet OM-2**
STRUCTURE	

PROCESS	
PEOPLE	Staff members involved: head of clinics, front office employee, surgeon, AIOS surgery, physician assistant, internist, AIOS internal medicine, clinical psychologist, dietitian, lab employee, and ECG employee.
RESOURCES	The resources that are used: 1. Information provided by the patient. 2. Lab values
KNOWLEDGE	-
CULTURE & POWER	A lot of protocols and principles are nationally the same, and apply to this hospital as well. Due to the nature of a hospital, everything is written down and done through protocol. Nothing is done off the books.

OM-5
Checklist for the feasibility decision document

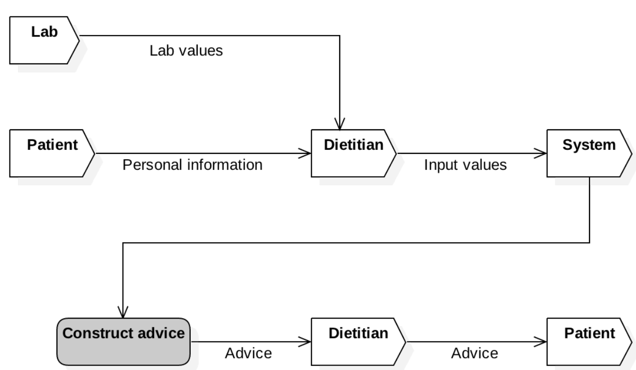
Organization Model	**Checklist for Feasibility Decision Document: Worksheet OM-5**
BUSINESS FEASIBILITY	1. The benefits for the organization are mostly economical. The dietitian doesn't have to manually come up with the needed values of a patient, and can just enter the values in the system, and come up with a diet according to those values. 2. According to the expert, the work load of the expert would decrease with 40-50%. The expert makes approximately 38.400 per year, with 11 dietitians working at the hospital. This means that, if the system would be used by all the dietitians with a decrease load of 40%, the system would theoretically make 168.960 euros. $(38.400 * 11) * 0.40 = 168.960$

	3. The expected cost of making the application is 360 work hours, according to 8 weeks of 20 hour work done, with 2 persons. 360 hours, with an hourly rate of 40 euros, means 14.400 euros. However, this is only a very basic application, made by students in a knowledge engineering course. If the application is made into an elaborate application, and implemented into hospitals, the costs will be a lot more, and cannot be properly calculated right now. 4. According to point 2, 168.960 euros per year is 40% of the spending every year by hospitals. And since there is no other alternative, the system introduces an interesting alternative. 5. Organizational changes are required, because the database management systems need to be altered. The administration needs to be rearranged as well. 6. Not only economic risks are present, but political and social risks as well. If a hospital fails to deliver, not only the revenue decreases. Patients need to be cured, and have a lot of different humanitarian rights. If the system fails to deliver, the consequences will be grand.
TECHNICAL FEASIBILITY	1. The system needs a lot of different rules, but can be made solely with if statements, and classes. 2. The quality needs to be superb, due to the nature of the problem. 3. It is clear what the success measure is, which is the correctness of the advice. 4. The user interface does not need to be complex. The user needs to fill in a lot of values, and the system needs to return other values. No need for a fancy user interface. 5. Initially, there is no communication with other systems. However, this could also be implemented (connection with the patient files, to store the values, for example). However, this is not mentioned by the expert as a necessity. 6. The technological risks, if connected to other systems of the hospital, are all the current systems within a hospital. All systems are connected with each other, through a cobweb of information flows, often with a number of flaws due to the sheer age of the systems. Due to the manual input of the user, this problem should not occur. However, if a system administrator links the system, one way or another, problems could occur.
Project feasibility	1. The expert thinks that the system could introduce a great advantage to a hospital. 2. The needed resources can be made available. 3. The required knowledge is present. 4. The expectations regarding the project and its results can be debated about. Because the hospital has a lot of (unknown to us) protocols and systems, the system could have a hard time being implemented into the hospital. However, on paper, the expectations are realistic. 5. The project organization is adequate. 6. There are no further project risks and uncertainties known as of now. However, we must keep our eyes open for potential problems along the way, because a hospital is a very delicate and large environment.
Proposed actions	1. 1. *Focus:* The focus needs to lay into the reliability of the system. A bad design of the diet values is the greatest risk. 2. *Target solution:* If made for a large scale, and not for a school project, a solid team of system designers and engineers needs to be appointed to develop the system.

	3. The result will be a cost efficient system that results in more time for dietitians to focus on important things. The costs could differentiate between 15.000 and tons. The estimated benefits will be around 168.960 euros per year. 4. Designers and engineers need to be fixed to begin with the project. 5. *Risks:* If the system fails after implementation, the consequences could be enormous.

TM-1
Refined description of the tasks within the target process

Task Model	**Task Analysis Worksheet TM-1**
TASK	Construct diet advice.
ORGANIZATION	The advice is constructed within the system. This system is used by the dietitian.
GOAL AND VALUE	When the system delivers accurate advice, without a notification for further research, the dietitian does not need to add something to the advice. If the system does not succeed, the dietitian need to design the advice themselves. Therefore: the added value is large.
DEPENDENCY AND FLOW	Below we can see the dataflow diagram associated with the construction of dietetic advice:



OBJECTS HANDLED	*Input objects*: The system needs the lab values of a patient, and some personal information. These are all knowledge items. *Output objects*: The system exports diet advice. This device can range from: ”The patient is doing great”, to: ”The patient needs to take more B12, and protein. The patient has a headache due to insufficient water intake. The patient loses too much weight, and needs to eat more.” *Internal objects* The system code, with its respective algorithms, formulas, and design structures. Furthermore, a number of rules need to be added.

TIMING AND CONTROL	The task takes a couple of seconds. The system does not need a lot of computer power for this task. Describe control constraints: (i) The preconditions of the system are power and an internet connection. (ii) The postconditions are power and an internet connection as well.
AGENTS	The dietitian uses the system. The system itself constructs the advice.
KNOWLEDGE AND COMPETENCE	See TM-2.
RESOURCES	- The dietitian works with the system. Therefore, staff time is used. - The system is homogeneous, and therefore does not use other systems. - The system needs maintenance and bug fixing if a bug occurs. All this costs money and staff hours.
QUALITY AND PERFORMANCE	The system needs to return correct information. If not, the patient will be at risk. The dietitian could double check the information on correctness. However, the advantage of the system will then be compromised. Concluding, the system needs to work flawlessly. This is not a problem due to the relatively simple rules, which are pretty fault-proof.

TM-2

Specification of the knowledge employed for a task, and possible bottlenecks and areas for improvement

Task Model	**Knowledge Item Worksheet TM-2**
NAME	Advice
POSSESSED BY	Knowledge system
USED IN	Construct advice
DOMAIN	Dietetic advice after stomach reduction

Nature of the knowledge		**Bottleneck / to be improved?**
Formal, rigorous		
Empirical, quantitative		X
Heuristic, rules of thumb		
Highly specialized, domain-specific		X
Experience-based		
Action-based		
Incomplete		

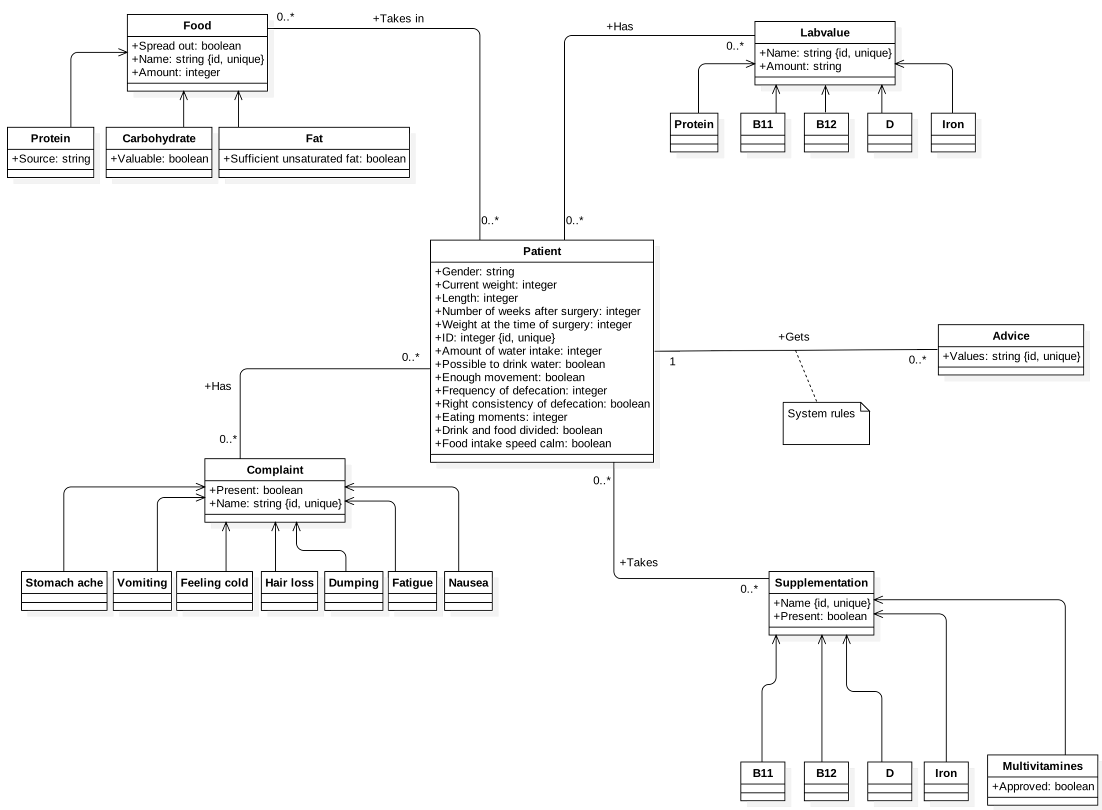
Uncertain, may be incorrect		X
Quickly changing		
Hard to verify		
Tacit, hard to transfer		
Form of the knowledge		
Mind		
Paper		
Electronic		
Action skill		
Other		
Availability of knowledge		
Limitations in time		X
Limitations in space		
Limitations in access		
Limitations in quality		
Limitations in form		

AM-1
Agent specification according to the CommonKADS agent model

Agent Model	**Agent Worksheet AM-1**
NAME	*Dietitian*
ORGANIZATION	The dietitian works on a specific dietetic department. Within this department there are a number of other employees, which all work side by side.
INVOLVED IN	Give patient advice and check values.
COMMUNICATES WITH	Surgeon, front office employee, physician assistant, internist, clinical psychologist, lab employee, and ECG employee
KNOWLEDGE	Lab values, patient values, and patient input.
OTHER COMPETENCES	Dietetic schooling and knowledge of the organization.
RESPONSIBILITIES AND CONSTRAINTS	The patient needs to do most of the work, but if the dietitian gives wrong advice, the patient will not move forward. There are no further constraints.

3 Knowledge Model

3.1 Domain Model



The knowledge circulating in our system can be described as categories and properties involved. For the patient knowledge, the general features (gender, current weight, height, weeks after surgery, weight at time of surgery), complaints (Stomach aches, nausea, vomiting, feeling cold, fatigue, hair loss, dumping), and lab values (Protein contents of blood, vitamin levels of B11, B12, D, and iron levels) can be described. Also part of the patient knowledge is the monitored nutrition of the weeks leading up to the meeting, revealing the protein, carb, fat, water, vitamin intakes per day, and knowledge about the movement, defecation, eating moments, eating speed, and separation of food and drinking.

```
(patient.weight.surgery - patient.current.weight) /
patient.weeks.after.surgery =< 0.8 OR > 1.5
CAUSES
advice = WEIGHT.ALARM;

patient.weight/patient.length  =< 27
CAUSES
advice = WEIGHT.ALARM;

protein-lab.value =< 60
MEANS
```

```
advice = MORE protein-food;

b11-lab.value =< 5,9
MEANS
advice = supplementation-b11.present = TRUE;

b12-lab.value =< 200 AND supplementation-b12.present = FALSE
MEANS
advice = supplementation-b12.present = TRUE;

b12-lab.value =< 200 AND supplementation-b12.present = TRUE
CAUSES
advice = SPECIALIST.ALARM;

d-lab.value =< 20
MEANS
advice = supplementation-d.present = TRUE;

iron-lab.value =< 14 AND patient.gender = MALE
MEANS
advice = supplementation-iron.present = TRUE;

iron-lab.value =< 10 AND patient.gender = FEMALE
MEANS
advice = supplementation-iron.present = TRUE;

protein-food =< (0.8 * patient.length * 27)
AND protein.source = ANIMAL
MEANS
advice = MORE food-protein.intake;

protein-food =< (1.2 * patient.length * 27)
AND protein.source = PLANT
MEANS
advice = MORE food-protein.intake;

protein.spread.out-food = FALSE
MEANS
advice = protein.spread.out-food = TRUE;

carbohydrate.spread.out-food = FALSE
MEANS
advice = carbohydrate.spread.out-food = TRUE;

fat.spread.out-food = FALSE
MEANS
advice = fat.spread.out-food = TRUE;

protein.source-food = PLANT
MEANS
advice = MORE protein.amount-food;

carbohydrate.valuable-food = FALSE
MEANS
```

```
advice = carbohydrate.valuable-food = TRUE;

fat.sufficient.unsaturated.fat-food = FALSE
MEANS
advice = fat.sufficient.unsaturated.fat-food = TRUE;

patient.possible.to.drink.water = FALSE
CAUSES
advice = WATER.ALARM;

supplementation-multivitamines.present = FALSE
MEANS
advice = supplementation-multivitamines.present = TRUE;

supplementation-b12.present = FALSE
MEANS
advice = supplementation-b12.present = TRUE;

patient.enough.movement = FALSE
MEANS
advice = patient.enough.movement = TRUE;

patient.frequency.of.defecation = < 2 OR > 14
CAUSES
DEFECATION.ALARM;

patient.eating.moments = < 6
MEANS
advice = patient.eating.moments > 6 AND < 11;

patient.drink.and.food.divided = FALSE
MEANS
advice = patient.drink.and.food.divided = TRUE;

patient.food.intake.speed.calm = FALSE
MEANS
advice = patient.food.intake.speed.calm = TRUE;

complaint-stomach.ache.present = TRUE
AND patient.food.intake.speed.calm = FALSE
MEANS
advice = patient.food.intake.speed.calm = TRUE

complaint-stomach.ache.present = TRUE
AND patient.eating.moments = < 6
MEANS
advice = patient.eating.moments = > 5

complaint-stomach.ache.present = TRUE
CAUSES
advice = SOURCE.ALARM

complaint-nausea.present = TRUE
AND patient.food.intake.speed.calm = FALSE
```

```
MEANS
advice = patient.food.intake.speed.calm = FALSE

complaint−nausea.present = TRUE
AND patient.eating.moments =< 6
MEANS
advice = patient.eating.moments => 5

complaint−nausea.present = TRUE
AND food−fat.amount => 50
MEANS
advice = food−fat.amount =< 50

complaint−nausea.present = TRUE
CAUSES
advice = SOURCE.ALARM

complaint−vomiting.present = TRUE
AND patient.drink.and.food.divided = FALSE
MEANS
advice = patient.drink.and.food.divided = TRUE

complaint−vomiting.present = TRUE
AND patient.food.intake.speed.calm = FALSE
MEANS
advice = patient.food.intake.speed.calm = FALSE

complaint−vomiting.present = TRUE
AND patient.eating.moments =< 6
MEANS
advice = patient.eating.moments => 5

complaint−vomiting.present = TRUE
CAUSES
advice = SOURCE.ALARM

complaint−feeling.cold.present = TRUE
OR complaint−fatigue.present = TRUE
AND ((patient.weight.surgery − patient.weight) /
patient.weeks.after.surgery) => 1.5
MEANS
advice = food.intake = MORE

complaint−feeling.cold.present = TRUE
OR complaint−fatigue.present = TRUE
CAUSES
advice = HEALTH.ALARM

complaint−hair.loss.present = TRUE AND protein−lab.value =< 60
MEANS
advice = MORE food−protein.intake

complaint−hair.loss.present = TRUE
AND supplementation−multivitamines.present = FALSE
```

```
MEANS
advice = supplementation-multivitamines.present = TRUE
complaint-hair.loss.present = TRUE
CAUSES
advice = HEALTH.ALARM

complaint-dumping.present = TRUE
AND food-carbohydrates.valuable = FALSE
MEANS
advice = food-carbohydrates.valuable = TRUE

complaint-dumping.present = TRUE
CAUSES
advice = HEALTH.ALARM
```

3.2 Ontologies

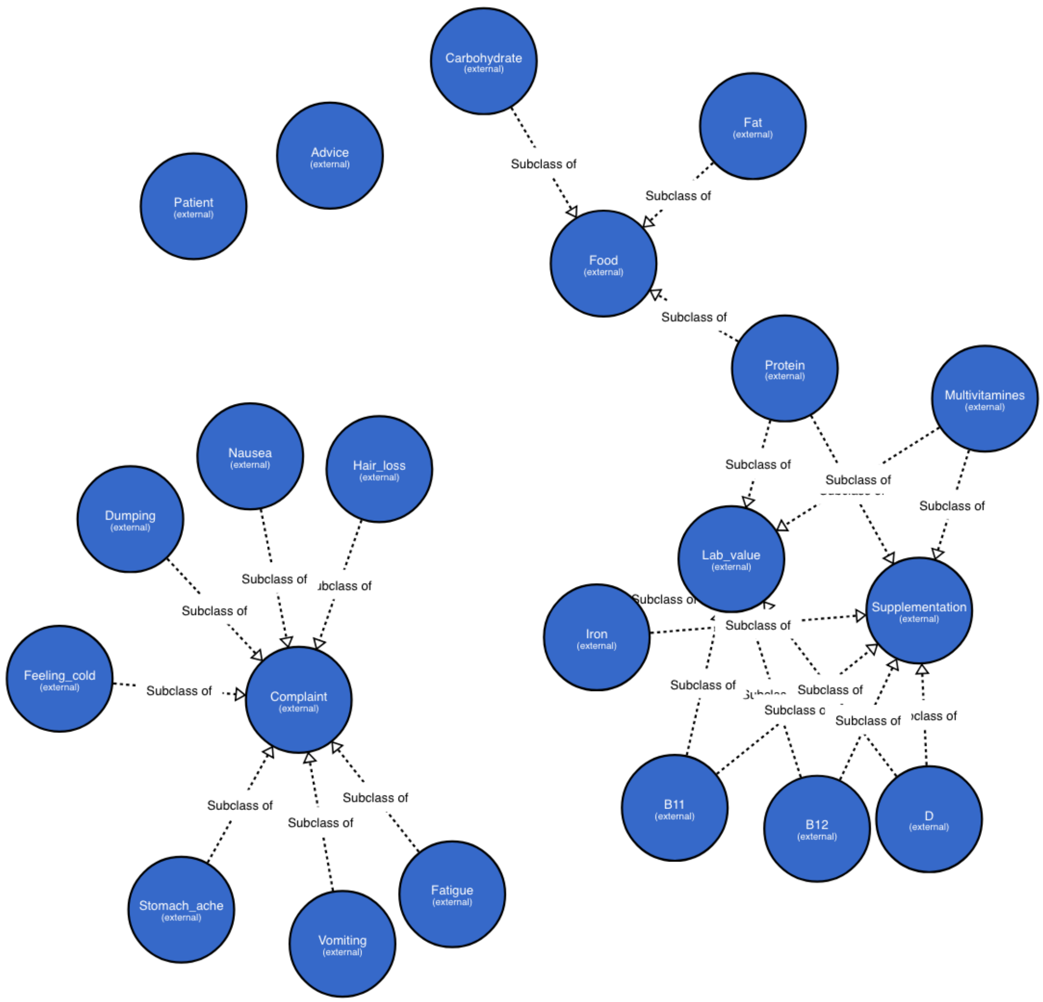
During the search for ontologies that might capture the knowledge contained in the system, several discoveries were made.

Food ontologies, such as BBC's Food Ontology (https://www.bbc.co.uk/ontologies/fo), tend to focus on the description of diets from a holistic viewpoint, where concepts such as a complete menu or a certain type of diet can be annotated. They generally don't focus on nutritional values, and as the system proposed matches certain nutritional non-quantifiable advices with input values, it becomes unimportant which actual diet is followed or which menus are used (the hows of accomplishing increased protein intake are left to the patient).

Nutritional ontologies or medical ontologies, such as the Ontology for Nutritional Studies (https://genesandnutrition.biomedcentral.com/articles/10.1186/s12263-018-0601-y), are complex in content and are focused on the academic field. In addition, they are not published online as a vocabulary to link to, and thus do not serve much more purpose than a self-made knowledge model, such as a class diagram.

Based on the small number of specific resources used by the niche system at hand, it was decided to just create an ontology based on the class diagram, where links to other ontologies or online vocabularies can be made within the ontology. The ontology itself, depicted below and included in the submission as an OWL file, can be used to annotate the code itself (documentation), or to annotate the outputs of the system (informational). The latter purpose results in a reuse potential between dietitians, a sharing capability between systems, and informational clarity for the patients themselves. For example, if the advice "Increase protein intake from animal-based food" is given, we can annotate the textual output in HTML for example, through Microdata, as follows:

```
<a itemscope
     itemtype="http://dieteticadviceontology.org/adviceX"
     href="http://dieteticadviceontology.org/adviceX" >
- Increase animal-based protein intake from food.
</a>
... repeat for all advice
```



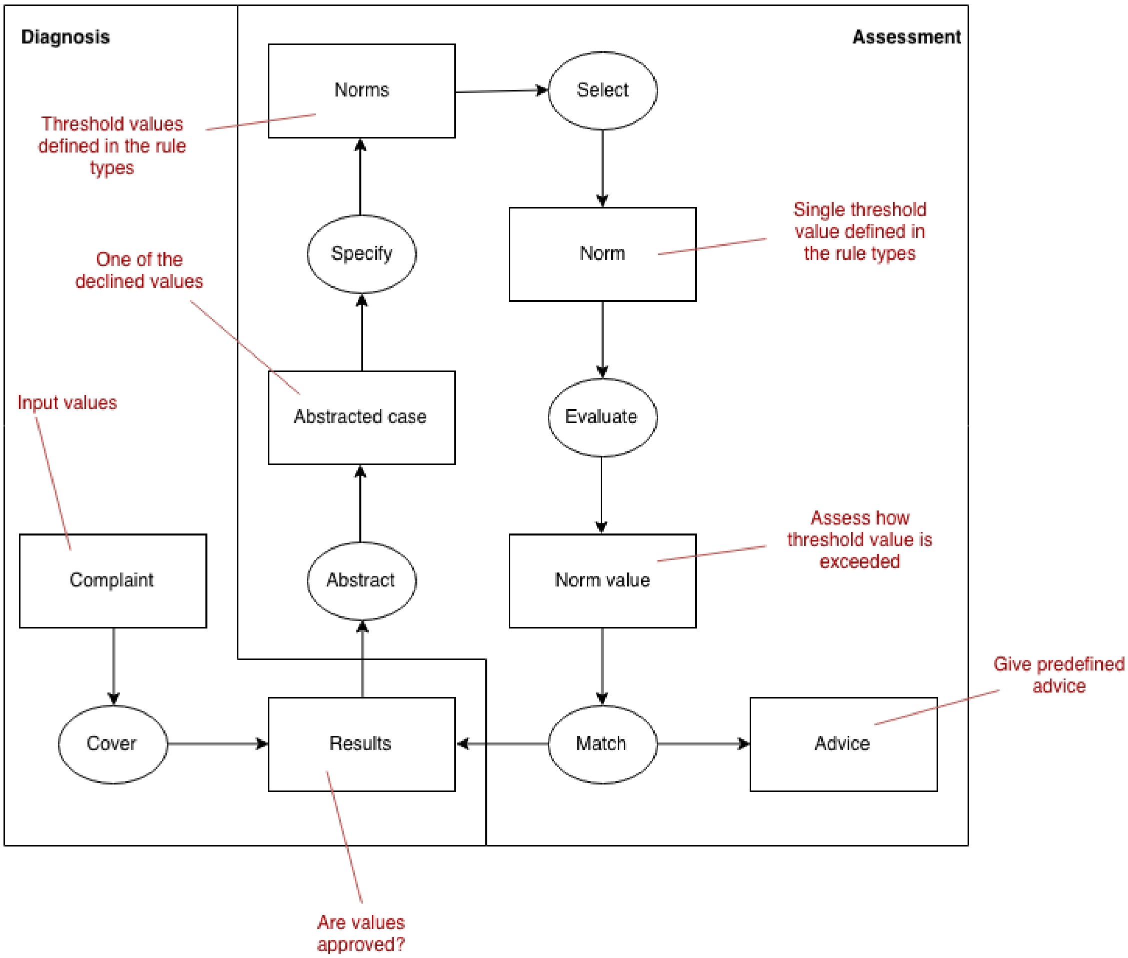
3.3 Inference Structure

All the threshold values are depicted in the domain model. What these threshold values mean can be seen in the inference structure.

The inference structure is depicted below. The structure is composed of two parts: the diagnosis part, and the assessment part. In the diagnosis part, all the input values are analyzed and checked whether they are approved by their respective threshold. For example:

```
input value: protein-lab.value = 40
threshold: 60
(input value < threshold)
... for all input values
```

Afterwards, only the disapproved values are further analyzed one by one. If the value exceeds a certain threshold, a specified advice is linked to the problem. The diagnosis template from the CommonKADS book is a lot more comprehensive. This template double checks other problems, to see if a hypothesis is true. However, due to the fact that we use predefined threshold values and boolean functions, this part is not necessary. The assessment template is fully used.



3.4 Task Knowledge

The task model is made with the help of the inference structure. Due to the fact that the inference structure is a combination between the diagnosis template and the assessment template, the task model is also a combination between the diagnosis template and the assessment template.

The task of dietetic advice construction during the second meeting after the surgery has been captured by the first interview. The task, which is a combination of a diagnosis and assessment task types, takes several input variables, finds the problems to solve, and maps solutions (pieces of advice) to the problems identified. The task model is depicted below. The inputs, rules, and outputs, are further described in the next section.

```
TASK diagnosis/assessment;
    ROLES:
        INPUT: case-description: "Personal values";
        OUTPUT: decision: "Dietetic advice";
    END TASK diagnosis/assessment;

TASK-METHOD diagnosis/assessment-with-abstraction;
    REALIZES: assessment;
    DECOMPOSITION:
        INFERENCES: cover, abstract, specify,
```

```
                select, evaluate, match;
        ROLES:
                INTERMEDIATE:
                        complaint: "The raw input values"
                        results: "Approved and disapproved values"
                        abstracted-case: "One of the declined values";
                        norms: "Threshold values defined in the rule
                        types";
                        norm: "A single threshold value";
                        norm-value: "How threshold value is exceeded";
                        advice: "Predefined advice";
        CONTROL-STRUCTURE:
                WHILE
                        NEEDS-ADVICE abstract(case-description ->
                        abstracted-case)
                DO
                        case-description := abstracted-case;
                END WHILE
                specify(abstracted-case -> norms);
                REPEAT
                        select(norms -> norm);
                        evaluate(abstracted-case + norm -> norm-value);
                        evaluation-results := norm-value ADD
                        evaluation-results;
                UNTIL
                        HAS-ADVICE match(evaluation-results ->
                        decision);
                END REPEAT
END TASK-METHOD diagnosis/assessment-with-abstraction;
```

3.5 Two Scenarios

In the subsequent sections, two scenarios are given, where the patient name and dietitian name is randomly assigned. These two scenarios were acquired by a second interview, in the shape of a system functionality workshop with the expert.

3.5.1 Scenario 1

James is a male patient of Sara, a dietitian who works at the OLVG in Amsterdam. James weighs 154kg, and has a height of 178cm. He has undergone a stomach reduction surgery one week ago. In the first week, James has only taken liquid foods, which he has done consistently. James has not given in to temptations, and does not have any physical complaints. After a brief talk during a meeting with Sara, Sara enters all the values of James into the system (lab values, all the food intake, etc.). Due to the fact that James has followed protocol, he has no complaints. The only advice given by the system is an increased protein intake, which makes sense, given the liquid nature of James his diet in the first week. James and Sara shake each others hand, and James leaves, encouraged that all his values are correct.

3.5.2 Scenario 2

Yasmin is a female patient of Ralph, a dietitian working at the TerGooi hospital in Hilversum. Yasmin has had a stomach reduction operation about two weeks ago. She weighs 143kg, and has a height of 164cm. During a standard meeting, Ralph finds out that Yasmin has a number of complaints. Since a couple of days, Yasmin has been dealing with nausea, stomach aches, and hair loss. Her protein value is 54 grams per liter, based on the lab results. Ralph listens carefully to Yasmin her story, and based on her input, Ralph enters everything in the system. The system returns a number of things:

- ”The hair loss is caused by a insufficient protein intake, increase protein intake through nutrition.”
- ”The patient does not separate food and water intake. Due to this fact, the patient feels nauseous and has stomach aches. The patient is advised to do so.”
- ”Iron blood value is insufficient. More nutrition, rich in iron, or additional iron supplementation is required.”
- ”The fat intake of the patient is too high, a lower fat intake is advised.”
- ”The patient does not exercise enough. The advice is to move 30 minutes to an hour, 5 times a week, with moderate intensity.”

The output is quite comprehensive. Yasmin needs to improve a number of things, to ensure a safe and healthy weight loss. The system has identified the most likely causes of the complaints mentioned by Yasmin. If the patient follows the advice of the system, these complaints should decrease. Yasmin asks the dietitian which food has a lot of iron, because she should take more iron. The dietitian suggests lean steak, which also does not have a lot of fat. With this advice, and the advice from the system, Yasmin leaves the hospital, hoping that her complaints will disappear when she alters her diet.

4 Communication Model

CM-1

Specifying the transactions that make up the dialogue between two agents in the Communication Model

Communication model	Transaction Description Worksheet CM-1
TRANSACTION IDENTIFIER/NAME	- Enter input values in the system: in this transaction, the dietitian knows the values, which he/she enters into the system. The system uses these values to analyze the patient and construct an advice. - Display dietetic advice: once the system has constructed the advice, the advice is shown to the user.
INFORMATION OBJECT	Core information object: patient values.
AGENTS INVOLVED	The dietitian enters the patient values, and the system receives them.
COMMUNICATION PLAN	Communication between dietitian and the system.
CONSTRAINTS	All the required values need to be known. If the, for example, protein value is not known, the system can not work.
INFORMATION EXCHANGE SPECIFICATION	The system asks about all the input values one by one.

5 Implementation

To implement the system, Python 3 (https://www.python.org/) was used. The system was designed as a command-line application where questions are asked, input variables are manually entered one by one, after which the dietitian will be presented with separate recommendations or advice, together shaping a complete advice for the respective second meeting after surgery completion. The complete code is provided in the submission of our project.

The implementation consists of two python scripts, of which one contains the class Patient—patient.py. The main program file contains the input acquiring functions and advice construction methods—system.py. For runnability of the system outside of a Python environment such as PyCharm (https://www.jetbrains.com/pycharm/), a combined python file, dieteticsystem.py, is provided which can be run from a command prompt.

The patient class contains all the variables gathered during the meeting with the dietitian in one class, contrasting the separate classes designed in the class diagram. The decision to do this resides in the single-property instances of intake values, lab values, and complaints. Additionally, the lab values and intake measures usually consist of simple booleans or singular values expressed in some metric unit. To create separate classes for this is programming overhead and does not really serve a purpose, as collecting all of these variables within one class is still manageable and eases the access to these values by the main program. Finally, the main program provides the patient class with the complete meeting advice given, and creates and stores an overview in a single text file related to the patient. The patient class includes the following functions:

```
def __init__(self, name, gender, height, weeks_after_surgery,
             weight_at_surgery, current_weight)
    ...

def enter_lab_values(self, lab_date, protein, B11, B12, D, iron)
```

```
    ...

def enter_complaints(self, stomach_ache, vomiting, feeling_cold,
            hair_loss, dumping, fatigue, nausea)
    ...

def enter_intake(self, water, water_possibility, supplements_B12,
            protein_intake, protein_source, protein_intake_spread,
            carb_intake_valuable, carb_intake_spread,
            unsat_fat_intake_sufficient, fat_intake_spread,
            supplements_multivitamins, fat_intake,
            enough_movement, defecation_freq,
            right_defecation_consistency,
            eating_moments, divided_food_water, slow_intake)
    ...

def create_overview(self, advice)
    ...
```

The system code encompasses the meeting communication functions which gather the information necessary from the patient. These correspond to the task components of diagnosis of personal values. The main program code also contains functions for processing the patient variables and constructing a readable, or textual, advice, to which end it uses the rules described before. To elaborate, every advice as of now consist of a single textual advice given (even though it can be semantically annotated in the output). These functions correspond with the control structure of the task described, and specifically the evaluation of all the input values through norms, until a complete advice has been constructed. The collection of advice is stored in a set, so that duplicate advice is disregarded. The main program code includes the following functions:

```
CONSTANTS AND ADVICE STRINGS

#program functions
def gather_general_info()
    ...

def gather_lab_values()
    ...

def gather_intake_information()
    ...

def gather_complaints()
    ...

def create_advice(patient:Patient)
    ...

def display_advice(general_advice, alarms, complaint_specific_advice)
    ...

THE INPUT PARSER
```

6 Methods and Reflection

6.1 Methods

The proposed system was designed based on two expert interviews. The first interview provided a holistic overview of the task at hand, the goal of the system, and specifics about domain-specific knowledge unavailable at the start. The second interview took the shape of the knowledge elicitation technique workshop, where we acted out two scenarios of patients visiting the expert, and thus were able to precisely observe every step taken during the dietetic meeting. The second interview also provided an opportunity to revise and add rules on the knowledge, which the system evaluates to construct an advice. The implementation includes all of the components described in the knowledge model —although in somewhat different form due to time limitations and convenience considerations.

6.2 Reflection

Overall, the system creation process was traversed well. The collaboration between the two team members was great and no extreme circumstances forced either to do more work than the other. A problematic point was the time-intensity of the CommonKADS models. It is our opinion that a number of steps in the CommonKADS are a bit superfluous, and sometimes a bit redundant. The CommonKADS way of designing a knowledge-based system seems to focus on the waterfall approach of software development, requiring a lot of defining and designing to be done before starting the actual implementation. We have taken a somewhat more agile approach to the project, in contrast to the waterfall method. We found this a more effective way to develop the system which we introduced in this report. Furthermore, the expert we consulted was very close to both of us. This made the expert contact fairly easy and the appointment scheduling easier than with somebody unknown.

6.2.1 Further Work

As the system as of now delivers a textual file as output, the storage of given advice (in a database, for example) and the concept of later retrieval of saved information needs to be explored further. The database management system to use could be explored, the implementation of it still needs to be done, and the Python application should be designed to either include the dietitian meeting stored knowledge. This way the created knowledge can be stored for reuse by other dietitians and can be referred to in further meetings. In addition to this, a graphical user interface for the system should be designed to make the interaction with the dietitian an easier, more efficient, process.

The knowledge components in the code and the output should be annotated with a link to an ontology or vocabulary describing the knowledge further. This way, every function will be well documented with regards to the inputs of the function and the returned values. In addition to the Python code, a database management system will be designed to store the patient values determined and the advice given. The ontology, could b e extended by linking resources to other online ontologies or vocabularies, so that the knowledge produced by the system can be shared. A web page dedicated to the proposed ontology and its implications for the patient and dietitian could be created to increase the informational value disseminated during the meetings and after.