NTNU
Norwegian University of
Science and Technology

# Integrating Case-based and Bayesian Reasoning for Decision Support

**Anne-Marit Gravem**

Master of Science in Computer Science

| | |
|---|---|
| Submission date: | June 2010 |
| Supervisor: | Agnar Aamodt, IDI |
| Co-supervisor: | Tore Bruland, IDI |
| | Helge Langseth, IDI |

Norwegian University of Science and Technology
Department of Computer and Information Science

# Problem Description

In this Master Project a method that combines CBR and BN, aimed to be used in pain classification, will be specified and developed. First, a literature study will be made, of CBR and BN combinations in medical diagnosis and treatment. A method will be designed that has CBR as its core reasoning approach and use BN to support inferencing in one or more steps of the CBR cycle. An experimental implementation will be made, based on existing CBR and BN tools within the TLCPC research group, i.e. jColibri and Smile/Genie, respectively.

Application of the method to case data and Bayesian models for palliative pain requires that these data and models become available in due time, as is the plan. In case of delays, and intermediate application will be selected, based on available suitable data and models from generally available repositories.

Assignment given: 15. January 2010
Supervisor: Agnar Aamodt, IDI

# Abstract

In this thesis, we present an approach to integration of case-based reasoning and Bayesian reasoning for decision support. Our design is meant to provide physicians with decision support in the context of palliative care for lung cancer patients. Because of delays in the medical data, we created an intermediate application with the aim to assist people in choosing an adequate wine for a given meal. We have developed a system that is able to utilize both the general knowledge of the Bayesian network and the specialized knowledge of the case base. Our results shows that the combination of CBR and BN are able to discover solutions that would not been found by using only one of the methodologies.

# Preface

This report constitutes my master thesis. It has been written and developed during my 10th semester of the Master of Science studies in Computer Science at the Norwegian University of Science and Technology (NTNU). This thesis is an continuation of the work done in course TDT4500 - Intelligent Systems, Specialization Project. This work have been carried out at the Department of Computer and Information Science (IDI) in the period between 15th of January and 25th of June 2010.

I would first like to thank my supervisor, Agnar Aamodt. I am extremely grateful for the honor to be working with him. He has been an invaluable source of knowledge and his feedback has increased the quality of my work.

I would also like to thank my co-supervisors, Tore Bruland and Helge Langseth for giving me advice and pointing me in the right direction when I was lost. Jorunn Hoøen, an wine expert at Vinmonopolet, has also been an important resource in my thesis, as she evaluated our system and gave us valuable feedback.

In addition, I would also like to thank my co-student, Kim Pedersen, which I have been working together with on the development of our system.

Last, but not least I would like to thank my family and friends for all the support they have given me the last five years, and especially my beloved daughter Hannah for being so understandable and bringing so much joy into my life.

To my grandpa, who past away this spring: Your pride meant a lot to me.

*Trondheim, June 25, 2010*

Anne-Marit Gravem

# Contents

**1 Introduction** **1**
1.1 Motivation . . . 1
1.1.1 Palliative Care . . . 2
1.1.2 The TLCPC Project . . . 3
1.1.3 Clinical Decision Support Systems . . . 4
1.2 Objectives of this thesis . . . 5
1.3 Shared work . . . 6
1.4 Overview of Report . . . 6

**2 Background** **9**
2.1 Case-Based Reasoning . . . 9
2.2 Bayesian Networks . . . 11
2.3 Earlier work . . . 13
2.4 Related systems . . . 14
2.4.1 Literature study results . . . 14
2.4.2 Compared to our design . . . 18

**3 Design** **21**
3.1 Choosing a Dataset . . . 21
3.2 Fundamental Design . . . 23
3.3 Technologies used . . . 24
3.3.1 jColibri . . . 24
3.3.2 GeNIe and SMILE . . . 26
3.3.3 Protégé . . . 26
3.3.4 myCBR . . . 28
3.3.5 Apache Derby . . . 29
3.3.6 Hibernate . . . 29

**4 Implementation** **31**
4.1 The Meal Case Base . . . 31

4.1.1 Data structures . . . . . . . . . . . . . . . . . . . . . . . . . . 32
4.2 The Meal BN . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34
4.3 System solution . . . . . . . . . . . . . . . . . . . . . . . . . . . 37
4.3.1 Configure . . . . . . . . . . . . . . . . . . . . . . . . . . . 40
4.3.2 Precycle and Query . . . . . . . . . . . . . . . . . . . . . . 40
4.3.3 Cycle . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41
4.4 Running Bacchus . . . . . . . . . . . . . . . . . . . . . . . . . . 43
4.4.1 Choosing a meal . . . . . . . . . . . . . . . . . . . . . . . 44
4.4.2 Defining the similarity . . . . . . . . . . . . . . . . . . . . 45
4.4.3 The results . . . . . . . . . . . . . . . . . . . . . . . . . . 46
4.4.4 The explanation . . . . . . . . . . . . . . . . . . . . . . . . 47

**5 Testing and results** **49**
5.1 Testing environment . . . . . . . . . . . . . . . . . . . . . . . . 49
5.2 Testing the BN . . . . . . . . . . . . . . . . . . . . . . . . . . . 50
5.2.1 First test . . . . . . . . . . . . . . . . . . . . . . . . . . . 50
5.2.2 Revision of BN . . . . . . . . . . . . . . . . . . . . . . . . 54
5.2.3 Second test . . . . . . . . . . . . . . . . . . . . . . . . . . 58
5.3 Testing Bacchus . . . . . . . . . . . . . . . . . . . . . . . . . . . 60

**6 Discussion** **63**
6.1 The strength of the BN . . . . . . . . . . . . . . . . . . . . . . . 63
6.2 The strength of the case base . . . . . . . . . . . . . . . . . . . . 65
6.3 The strength of the system . . . . . . . . . . . . . . . . . . . . . 66
6.3.1 Similarity Functions . . . . . . . . . . . . . . . . . . . . . 68
6.4 Choice of technology . . . . . . . . . . . . . . . . . . . . . . . . 68
6.5 TLCPC and our design . . . . . . . . . . . . . . . . . . . . . . . 69
6.6 Further work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 70

**7 Conclusion** **71**

**REFERENCES** **73**

**Appendices** **79**

**A Similarity Functions** **79**

**B Wine terms** **82**

**C Test tables** **83**

# Chapter 1

# Introduction

Decision support is a widely studied research area in artificial intelligence [6, 13, 31, 33, 36, 40, 46]. A primary objective has been to find appropriate technologies to use in different applications. In this thesis, we want to explore the combination of Case-based reasoning and Bayesian networks for decision support, preferably in the medical domain. This work is a continuation of the work done in our specialization project, where we studied existing decision support systems with focus on combinations with CBR. The goal of the preliminary project was to get an general view of the different systems and to use this knowledge to propose an architecture that facilitates medical decision support with use of CBR and BN. In this thesis, we continue this work by implementing a decision support system based on the proposed architecture.

First, we will describe the motivation of this thesis, being the need for palliative care, the TLCPC project and clinical decision support systems. The objective of this thesis is summarized in 1.2. Some material presented was created in collaboration with Kim Pedersen, and the work distribution is described in section 1.3. The last section of this chapter gives an overview of this report.

## 1.1 Motivation

Over 10.000 people die from cancer every year in Norway. Lung cancer is by now the fastest growing cause of death together with other smoke-related diseases as chronic obstructive pulmonary disease (COPD or KOLS in Norwegian). These diseases leads to a lot of pain and there is a great need for better and more adaptive palliative treatment (see section 1.1.1). Figure 1.1 shows the progress in death causes for women from 1990 to 2008.

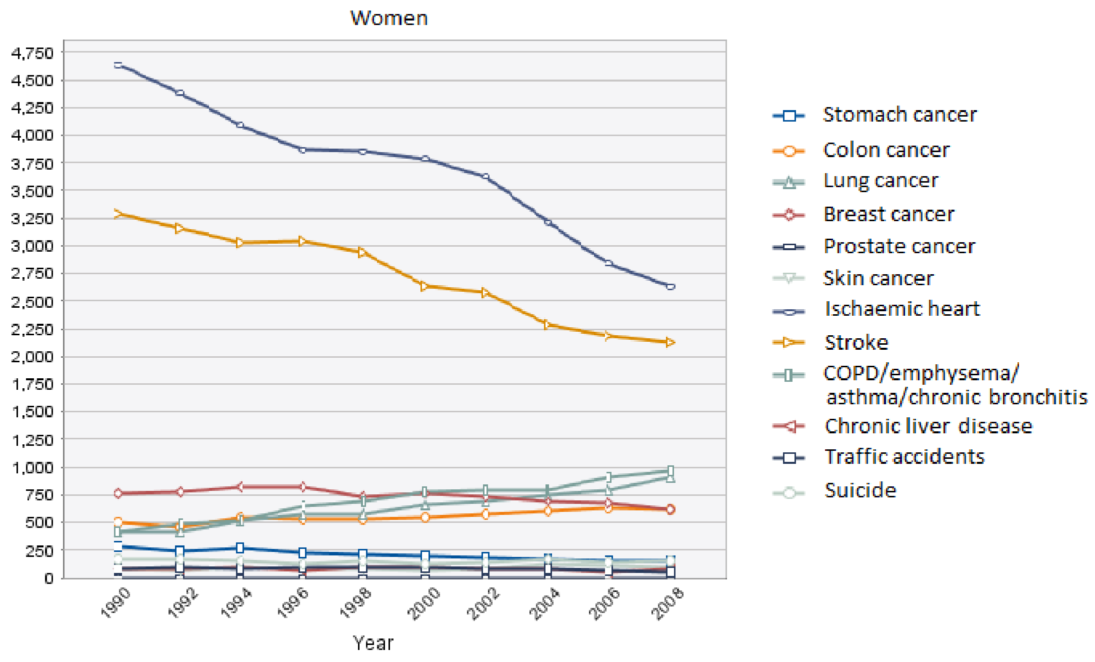


Figure 1.1: Causes of deaths for women from 1990-2008. Adapted from the Norwegian Institute of Public Health (http://www.norgeshelsa.no/).

Generating adapted treatments is complicated and time-consuming because of the need for experts on palliative treatment to specialize each patient treatment. There is reason to believe that this process can be done better and faster by introducing artificial intelligence techniques as Case-based reasoning and Bayesian networks. These techniques have, as far as we know, never been used together in medical decision support systems yet, and it is interesting to see how such an integration could work.

These next two sections is based on the work done in the specialization project [18] and is included for completeness and for a reader unfamiliar with the preliminary work.

### 1.1.1 Palliative Care

As we can see from figure 1.1, the number of women dying from lung cancer has increased in the last two decades. From being under 500 in 1980, the number has increased to approximately 1000 deaths in 2008. People dying from lung cancer have to depend on different sorts of treatment to maintain and improve their quality of life. This type of treatment is named palliative care. According to Kaasa and De Conno [24], the World Health Organization (WHO) has defined palliative care as follows:

> "the active, total care of patients whose disease is not responsive to curative treatment. Control of pain, of other symptoms, and of psychological, social and spiritual problems is paramount. The goal of palliative care is achievement of the best quality of life for patients and their families. Many aspects of palliative care are also applicable earlier in the course of the illness in conjunction with anti-cancer treatment"

Palliative care is a complicated field, and much research aims at finding an optimized way to treat patients in their last days. In 1988, the European Association of Palliative Care (EAPC) was established as a research network to create recommendations in treatments. This research network has established connections between the nations and published recommendations for palliative care for European countries [11, 19, 30, 43].

The EAPC research network fostered an EU founded project named the European Palliative Care Research Collaborative (EPCRC). One of the aims is to develop a computerized symptom assessment tool for palliative care. In close relation to this project, a Norwegian project called The Translational Research in Lung Cancer and Palliative Care (TLCPC) was initiated.

### 1.1.2 The TLCPC Project

The Translational Research in Lung Cancer and Palliative Care (TLCPC) Project [41] was founded by the Research Council of Norway(NFR) and focuses on pain treatment in cancer patients.

The TLCPC project aims to improve the treatment of pain in cancer patients by using proactive, advice-giving systems. Lung cancer is a common disease, and the survival rate is poor, only 10 - 15% survive for five years. Operations can cure lung cancer, but only 18 - 20% of the patients are applicable for operation. For the rest, the disease is detected too late. Patients with advanced lung cancer suffer from several symptoms and a general program for palliative care might help in meeting the patients needs [23].

Lung cancer was chosen as a research object in the TLCPC project because of its commonness, high death-rate, short disease course, high level of symptomatology and for the research group's earlier work and experience in the area [41]. The TL-

CPC project consists of six tracks, where four of them are research oriented and the last two deal with establishing the Central Norway Lung Cancer Biobank(CNLCB) and an international education program for PhD/Post-docs. The four research oriented tracks are genotyping (track A), gene expression (track B), computer based symptom assessment and classification (track C) and a decision support system (track D). The model is illustrated in 1.2.

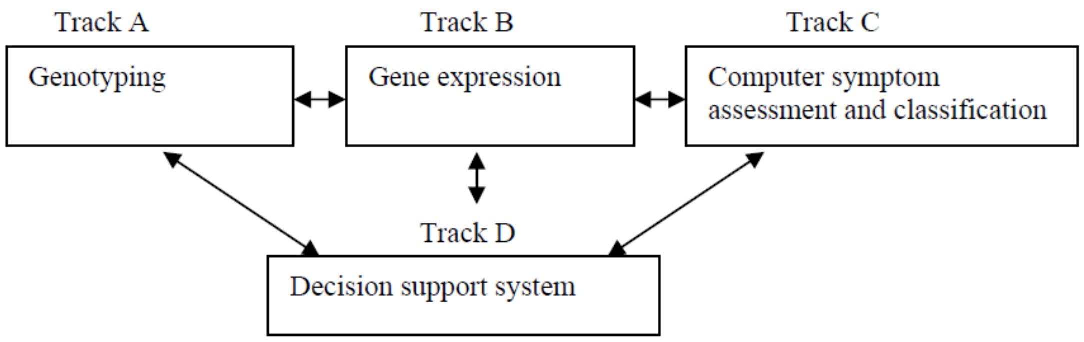


Figure 1.2: Model of the collaborations between the tracks A, B, C and D [41].

The EPCRC team (mentioned in 1.1.1) will contribute to the TLCPC project with scientific input on symptom assessment and classification into track D. Track D is also the main target for the study presented in this report. In the initial project description document [41], there were 3 main research questions to address in the TLCPC project:

1. How to best utilize the knowledge and information from the other tracks in an integrated decision support system for clinical practice?
2. How to combine generalized and situation specific knowledge to obtain the intended support?
3. Could such a system have any significance in clinical decision making?

This master thesis will mainly focus on the second issue by implementing and evaluating the architecture for clinical decision support that we proposed in our previous specialization project Gravem [18].

### 1.1.3 Clinical Decision Support Systems

Clinical decision support systems (CDSSs) are systems designed to assist medical personnel with decision-making tasks. Wyatt and Spiegelhalter [56] has defined them as:

> "active knowledge systems which use two or more items of patient data to generate case-specific advice"

The systems work by matching patient data to a computerized knowledge base in order to generate medical advice. Medical personnel and patients themselves can either manually enter data into the system, or the system could automatically collect the necessary data itself from electronic medical records.

A decision support system consists mainly of a knowledge base, an inference engine and a user interface. The knowledge base represents all the knowledge from cases, clinical guidelines and decisional rules, while the inference engine is responsible for processing the information and giving decision support. CDSS can support several different type of medical support; alerts, reminders, advice, critique and support [16].

Several studies have shown that CDSSs can improve physicians performance, but the effect on the patients outcome is poorly studied [16, 21, 22]. It has also turned out to be difficult to find a way to solve the problems coming from the increased complexity of the health care domain [40]. Montani [36] listed three general limitations regarding using the AI methodology case-based reasoning (CBR) in medical systems:

- Case data is becoming more complex, and this makes feature mining more complex also.
- Competence gaps (i.e. missing information in the case base) may lead to misleading information about how to solve the problem.
- Adaptation (i.e. to adapt retrieved solutions to solve a new problem) is challenging and time consuming.

The second and the third of these limitations can be solved by introducing Bayesian Networks (BNs) in the systems. BNs can help with both covering the competence gaps when the case base is incomplete and they can be used to adapt solutions. This way, we hope that the combination of CBR and BN can overcome former difficulties and create useful systems. Both CBR and BNs will be elaborated in sections 2.1 and 2.2.

## 1.2 Objectives of this thesis

The goals of this project are:

- To further specialize and implement an architecture that use a combination of CBR and BN for decision support.
- To evaluate the results of the implementation and its utility value.

## 1.3 Shared work

This work have been done in collaboration with Kim Pedersen. This section will describe the shared work.

First, I developed the first model of the system, integrating CBR and BN by using jColibri and SMILE. The design of the BN was then extended by Pedersen (see section 4.2) in order to add more information to the model at the same time as the probability tables were simplified. We also cooperated on creating a similarity function in myCBR (see appendix A). While my focus have been on developing the system and testing the proposed integration of CBR and BN, Pedersen have focused on creating explanations for the solutions generated. Hence, the explanations (see section 4.4.4) are developed by Pedersen and are not further explained in my thesis.

Our theses do overlap to some extent, but they have been written individually except from the section that gives a detailed explanation of our system (section 4.4).

## 1.4 Overview of Report

This thesis is meant to be read as a whole from beginning to end. Later chapters may refer to previous chapters. The thesis have extensively used color illustrations, and should therefore be printed in colors in order for the reader to get the most of the illustrations.

The next chapter describes the background material for this thesis, being the technologies, the associated NFR project and some related systems. Chapter 3 introduces the design of our system. Section 3.1 will describe the process of finding an adequate dataset, before the next section introduces the fundamental design based on the chosen dataset. The last section in chapter 3 covers the various technologies used in the development of our system.

Chapter 4 describes the implementation of our system. Section 4.1 and 4.2 addresses the meal case base and the Bayesian network. The next section, section

4.1.1, explains our data structures. Then, in section 4.3, the implementation of our system is reviewed. The last section in chapter 4 gives a detailed example of how the system is running.

In chapter 5, the testing of our system is described. The testing environment is addressed in section 5.1. First, the BN was tested separately (section 5.2), before the whole system were tested (section 5.3).

Next, in chapter 6, the results from our work are summarized and discussed. The first three sections deals with the strengths of the BN, the case base and the system respectively. In the next section, section 6.4, we discuss our choice of technology. Then, in section 6.5, we describe how our implementation can be used in the TLCPC project. The last section describes some further work that should be done.

Finally, chapter 6 concludes the work done in this thesis.

The appendices have 3 chapters. The first chapter lists the functions of the similarity functions built-in in jColibri. The next chapter describes some wine terms used in this thesis. The last chapter in the appendices lists the test tables from the tests in chapter 5.

# Chapter 2

# Background

This chapter presents the background of our research area. First, section 2.1 and 2.2 will respectively give an introduction to the AI methodology Case-based reasoning (CBR) and to Bayesian Networks (BNs). CBR was proposed as an methodology by the TLCPC project team because of its ability to reason about past situations [41]. In addition, BNs, a model that represents probabilistic relationships between features, was suggested to capture the general knowledge. In our previous specialization project [18], an architecture using these AI methodologies to support clinical decision support was proposed. This architecture is summarized in section 2.3. At last, section 2.4 will describe some systems that have been used to solve similar problems. This chapter is a further elaboration of the work done in the previous specialization project [18]. These systems will also be compared to our proposed architecture.

## 2.1 Case-Based Reasoning

Case-based reasoning (CBR) is an AI approach that solve problems based on experience from other similar situations. CBR is based on the human cognitive model, and is meant to mimic our way to predict and comprehend. It has gained broadly acceptance because of its psychological plausibility, and has been used in many successful applications [4, 10, 26, 34].

A case typically consists of three basic values; the problem description, the case solution and the reported result of applying the case solution. CBR is usually performed in a cycle of four main steps; retrieve, reuse, revise and retain (see figure 2.1). When a case to be solved enters the cycle, the CBR-system starts by retrieving the most similar case (or set of cases). The information in the case(s) is then reused to solve the problem. The solution is revised to see if it did solve the new problem, and then the useful part(s) of the solution is retained to help solve later problems.

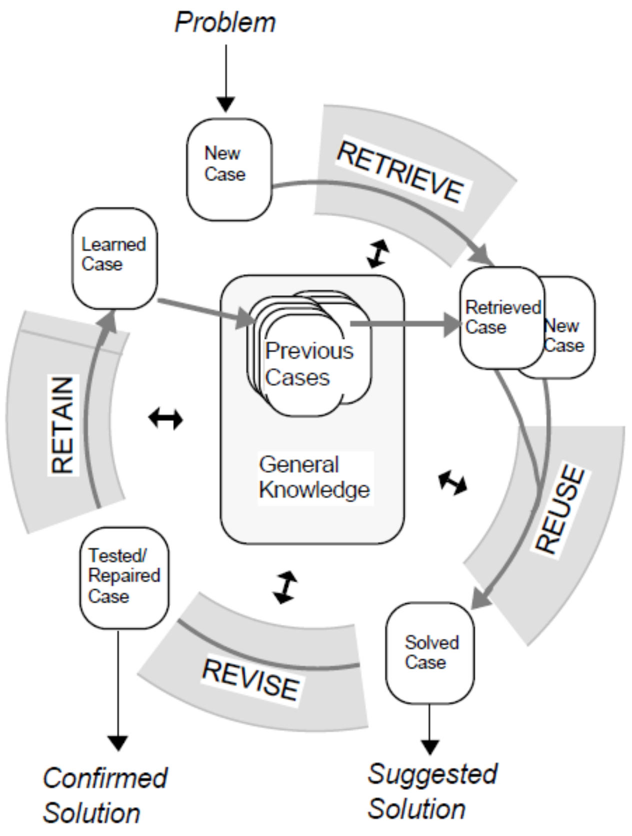


Figure 2.1: The CBR cycle developed by Aamodt and Plaza [2]

This is very much the same as how people tend to solve problems. For example, when a doctor gets a new patient, he will likely try to remember patients with the same symptoms. He then tries to treat the new patient based on his earlier experience, and he categorizes the outcome of the treatment in his mind. If the patient got better, he will remember it as a good plan, or if the patient got worse, he will remember it as a bad plan. Also, if the doctor ran some unnecessary test or if something unusual happened, he will remember these experiences the next time he gets a similar patient.

Because of this resemblance with the way physicians reason, CBR has been used to assist physicians in clinical decision support systems, of which an overview was given in 1.1.3. CBR has also shown to be a good methodology to combine with other reasoning approaches, and a summary of the work done can be found in Marling et al. [32]. As mentioned, the TLCPC project team wants to combine CBR and BN in a clinical decision support system (CDSS). As far as we know, there exists no CDSS where CBR and BN have been combined, but they have been successfully combined in other systems [12, 13, 28, 31].

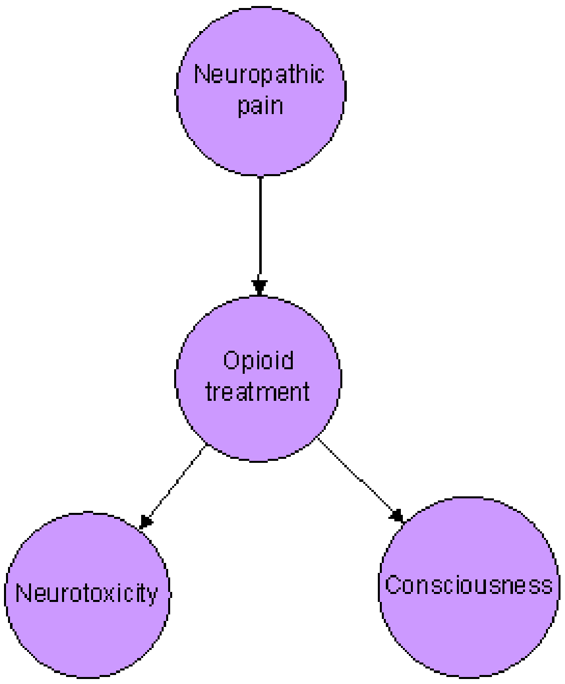
## 2.2 Bayesian Networks

A Bayesian network is a data structure used to represent dependencies between stochastic variables. The network is represented as a directed graph where nodes have conditional probabilities. Russell and Norvig [47, p. 493] formally specifies a Bayesian network as:

- A set of random variables makes up the nodes in the network. Variables may be discrete or continuous.
- A set of directed links or arrows connects pairs of nodes. If there is an arrow from node X to node Y, X is said to be the parent of Y.
- Each node $X_i$ has a conditional probability distribution $P(X_i|Parents(X_i))$ that quantifies the effect of the parents on the node.
- The graph has no directed cycles (and hence is a directed, acyclic graph, or DAG).

A sample of a Bayesian network and one of the nodes probability table is illustrated in Figure 2.2a. The topology of the network represent the relationships in the specified domain. Nodes represent the variables (e.g., `Neuropathic pain`), and the links represent the influences. As we can see in the figure, `neuropathic pain` influence treatment with opioids, and `opioid treatment` influence the `consciousness` of the patient. The figure also shows that `consciousness` and `neurotoxity` is conditionally independent given `opioid treatment`. Table 2.2b shows an imagined example of a probability table for node `consciousness`. The node is dependent on `opioid treatment`, so this node is represented in the table. As we can see, if the patient had a small dosage of opioids, the probability for the patient being barely conscious is 1 %, while is the dosage was strong, the probability is 80 %. As we can see from the table, the result of `consciousness` in dependent on `opioid treatment`.

BNs have several qualities that can be useful in our system; it has a strong statistical basis, it has a symbolic representation in the dependency network and its relations in the network may be given semantic interpretations (such as causality)



(a) An example of a Bayesian network.

| **Opioid Treatment** | **Consciousness** | |
|---|---|---|
| | **Barely** | **Regular** |
| **Small dosage** | 0.01 | 0.99 |
| **Normal dosage** | 0.20 | 0.80 |
| **Strong dosage** | 0.80 | 0.20 |

(b) An example of the fictitious probabilities of the node `Conciousness`.

[1]. Bayesian networks have been used to create medical models before, as Lucas [29] causal model of aortic coarctation. The Bayesian network is meant to be used to represent dependencies between state-variables in the clinical decision support system.

## 2.3 Earlier work

A literature study of clinical decision support systems was performed in the previous specialization project [18]. In this study, we found no existing systems which make use of both case-based reasoning and Bayesian networks in the medical domain. Based on the knowledge gained by the literature search, we developed a high-level architecture for clinical decision support. This architecture combined clinical guidelines (i.e., formal instructions for medical treatment), case-based reasoning and Bayesian networks to get the desired knowledge support. The architecture can be found in figure 2.2.

We imagine that every patient has his own electronic patient profile, where all his/hers information is saved. When a patient receiving palliative treatment is feeling unwell, a patient problem (e.g., a patient experiencing pain and feeling unconscious) is created. Our proposed architecture starts with sending a patient problem through the clinical guidelines. If the guidelines match the patient profile, a treatment is generated right away. If not, information from the last complying step (e.g., opioid dosage recommendations) is used for further reasoning together with any other relevant information from the patient case. This information is used to create a new case. The features of the new case is used to retrieve similar cases from the knowledge base. Features are connected to the Bayesian network by a feature link (marked by the dotted line between a node in the Bayesian network and a case). This way, the Bayesian network can be used to select the most similar cases, in order for them to be retrieved. The retrieved cases can then be adapted to solve the new case by utilizing the structure of the Bayesian network. The proposed solution is sent to a physician for revision. The physician can either approve the solution, or he/she can mark it as non-valid. If the solution is marked as non-valid, the physician is prompted to correct the solution (and thereby creating a valid solution). Both valid and non-valid solution can be saved by the system in order to correct and update the assumptions that created the case solution. A valid solution is sent as a suggested therapy to the patient and the BN is then used to justify if the solution should be added to the case base. Finally, the outcome of the treatment is used to update and correct the BN.

A more thorough examination of the architecture can be found in [18].

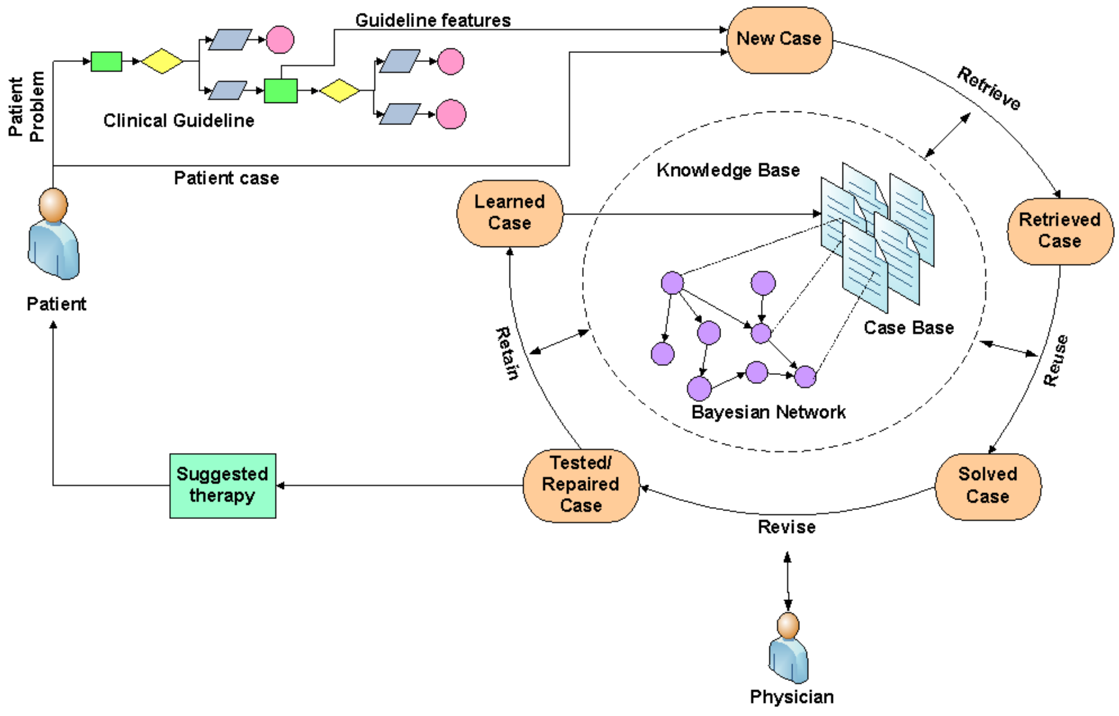


Figure 2.2: The suggested architecture

This architecture will be further exemplified and developed in this master thesis. We will specify a detailed architecture that is supported by the selected CBR an BN tools. This architecture will be implemented in order to evaluate its strengths and benefits.

## 2.4 Related systems

In this section, we will describe some related systems and compare these systems to our design.

### 2.4.1 Literature study results

In the specialization project [18], we did a literature study were we found some interesting systems dealing with clinical decision support. This section elaborates the work done in the previous specialization project.

Rossille et al. [46] have created a decision support system to support treatment of cancer patients (with main focus on breast cancer). Their system is meant to be a data warehouse in oncology by storing valuable information used in the treatment of patients. The system uses rule-based reasoning (RBR) to reason with guidelines, and CBR to reason with cases. CBR is here the secondary technology, used when a medical case is not compatible with the guidelines. When cases are stored in the case base, they are compared to an appropriate guideline. A case is compared to the steps in the guideline, and the last step that complied to the case is saved in a classification table that holds the cases and the guidelines. This makes it possible to identify cases that belongs to a specific guideline step.

Another relevant system is the one recently created by Marling et al. [33]. Their CBR system aims to help diabetes patients improve their insulin pump therapy. The system differs from other CBR applications in that it has to find the problem to solve itself. The system searches the database for problems encountered with a specific patient, and displays the problems it found. The user then select one of the problems and the system compares it to the case base. A k-NN algorithm is used to retrieve the cases, and only those that are above a given similarity-score are displayed to the user. The user decides if the cases will be used as a advice to the patients, and if so, in what form.

Montani et al. also created a system for managing diabetes with multi modal reasoning, named telematic management of insulin dependent diabetes mellitus (T-IDDM) [37–40]. Their system started out as a multi-modal reasoning system using rule-based reasoning (RBR) and CBR. In their system, the cases are classified (using a Naive Bayes strategy), creating a library of categorized prototypical cases. This classification is used to retrieve similar cases. In addition, cases are chained together with the previous and following ones, making it easy to retrieve the whole patient's history and to see the progress or any transitions between the classes. RBR is the main reasoning methodology, and it provided insulin dosage recommendations. The system recorded periodical control visits, and as the case library grew, CBR was used to specialize and adapt the rules on the basis of the patient's characteristics and its experience. The medical personnel using the system can choose if the CBR should retrieve cases only from the most probable class, or if they should be chosen from a set of probable classes. Also, the physician has to analyze and approve a case solution before it is stored in the case base. This way, they can be sure that the cases are quality assured by an expert, and not wild guesses by a system. Montani et al. added in [40] a probabilistic model of the glucose-insulin system, to cope with the problem of RBR-solutions being to general. This model-based approach was co-ordinated with CBR in the therapy revision phase, where they were used to specialize the rules in a mutually exclusive way.

The two diabetes systems ([33] and [40]) differ in their form of therapy. Marling et al.'s system use insulin pump therapy, which has to account for daily variations in diet and lifestyle. Montani et al.'s system instead uses classic insulin therapy, making these things unimportant. This is also the reason for the lesser role of CBR in the latter system.

Elvidge [14] recently presented a CDSS used for palliative care for cancer patients. The system is primary CBR-based, but it uses decision trees and the nearest neighbor algorithm to retrieve cases. Elvidge use care plans and guidelines to create simulated case solutions, this way representing general knowledge as cases. This is in contrast to the most of the other systems mentioned here, which mainly use RBR to represent general knowledge [37–40, 46, 49, 50].

CARE-PARTNER [6–8] is a multi-modal system used in long-term follow up of cancer patients who have undergone stem-cell transplantation. The system was created by Bichindaritz et al. and uses a combination of CBR, RBR and information retrieval (IR). CARE-PARTNER has several types of knowledge; monographs, scientific literature, practice guidelines, practice pathways and practice cases. While CBR reasons with the pathways and the cases, RBR reasons from guidelines and pathways represented by rules. The knowledge entities are represented in a network as nodes, and the links connecting them are defined by the Unified Medical Language System (UMLS), which provides an ontology for the medical domain. The system has several areas of application, and the first step of problem solving is to categorize the problem. It can be categorized as an information retrieval task, a problem-solving task or something else.

The problem-solving reasoning in CARE-PARTNER proceeds in six steps:

1. The system first creates an interpretation of the initial situation in the knowledge representation.
2. It searches the knowledge base to find suitable rules, pathways and cases. These searches are done in parallel.
3. This step resolves conflicts by using the entity (rules, pathways or cases) that has the most problem description elements that matches. If the entities have a matching number of problem description elements, a prioritized order are used. Rules are preferred over pathways, and pathways are preferred over cases.
4. The selected entity is reused to solve the problem. If it is a rule, it is fired, cases are adapted and pathways are used either with or without adaption.

Either the problem is solved, or new elements are added to the problem description.

5. Here, the case is updated and the knowledge representation elements that were used are marked. The problems that were solved are removed from the case, and if all problems are solved, the solution is proposed to the user. If not, the reasoning cycle restarts at step 2.

6. This step memorizes the complete solution and the target case.

Bichindaritz et al.'s system also has a way of revising solutions, by making it possible for users to add positive and negative feedback. Positive feedback from a user results in a validation mark of the memorized case, while the negative feedback gives a non-valid mark on the case. If a case is marked as non-valid, experts can give the system a better solution and the system can learn from its mistakes by studying the differences.

In CARE-PARTNER, all reasoning methodologies are equally important. The system aims to achieve a close cooperation by separating their reasoning steps and allowing for them to run in parallel (like step 2). Also, partial results from the reasoning methodologies can be used in the different reasoning cycles. CARE-PARTNER has been tested in [6], where it was shown to produce 98.6% adequate decisions on 163 different clinical situations and cases. This result shows that the system has managed to get very trustworthy. However, because of the strict rules in clinical practices, the authors states that the system needs to attain a rate of 100% adequate results to be used in routine clinical practice.

Researchers have also tried to incorporate Bayesian networks in CDSSs in different ways. Mani and Pazzani [31] suggest to generate clinical guidelines by using decision tables. In their approach, they used the BN framework to induce decision tables. They induced four different kinds of decision tables, and compared their accuracy on classifying clinical dementia rating scores. Bayesian ideas has also been used to developed clinical guidelines by both Landrum and Normand [28] and Diamond and Kaul [12].

Dingsoyr [13] created a framework for integrating CBR and BN for decision support. His framework uses BNs to compute similarity metrics and thereby help the CBR engine to chose the most similar case when no exact match is found.

CBR has also been combined with Bayesian reasoning in other domains. Tran and Schönwälder [52] presents a way to use probabilistic reasoning to support fault resolution in case-based reasoning. They propose to build a probabilistic model based on a partial case base. Their probabilistic reasoning method consists of two processes: a ranking process that narrows down the scope of the problem and a selection process that finds a solution to the problem by evaluating the correlation between cases and the fault.

Rodriguez et al. [45] used BN to create an probabilistic model for case-based reasoning. Their model utilized two Bayesian networks; one for ranking categories and another for identifying exemplars within the categories. They propose an exemplar-based model, which means that they only store prototypical cases. When a new case is given to the system, it has to first determine which exemplar in the case base that best classifies the case, and then if the new instance can be used to improve the accuracy of the model. Both BNs are updated if the new case is saved.

### 2.4.2 Compared to our design

These systems differ from our proposed architecture in several ways. Of the 5 systems combining CBR and RBR, two of them has CBR as the secondary technology with RBR as the main technology [40, 46], two has CBR as either the only or the main technology [14, 33] and one system uses CBR, RBR and information retrieval (IR) in parallel [6]. Our system is based on CBR, and uses BN as a secondary technology. We found very few publications dealing with use of BN in clinical decision support systems. In these articles, the researchers have used BN to develop clinical guidelines [12, 28, 31].

As opposed to this, we use the BN to represent dependencies between state variables. In our architecture, we also included clinical guidelines. These guidelines were not the main focus area for our specialization project and master thesis as the TLCPC project team are not done with settling these. The representation of these guidelines are therefore left as further work.

Dingsoyr [13]s approach is more similar to our. In his system, the Bayesian network is used to calculate the most probable case. The difference is that in his system, the BN is built from the case base. In this thesis, we want to test if having a network that represents generalized expert knowledge, while the case base is made up of patient cases, can strengthen our reasoning and produce good results. Most of the other systems we studied did also use the case base to create the BN [13, 20, 45, 52].

The role of the BN varies in the different systems. Several of the system used BN to develop clinical guidelines in CDSSs [12, 28, 31]. BN has also been used to fault reasoning [52], and for ranking and indexing [45]. None of these approaches are similar to ours. We want to use the BN as part of the reasoning, by using it to find attributes to look for in the cases that is retrieved.

As far as we know, our approach gives the BN a more apparent role in the reasoning process than what have been done before.

# Chapter 3

# Design

This chapter will give the reader an overview of the design of our system. In order to specialize the proposed architecture, we first have to chose a dataset to use. This dataset is essential for the development of a final architecture of our system, as it will affect how the case base and the Bayesian network is represented. The first section describes the process of finding an adequate dataset. Then, a fundamental design that is based on the dataset is presented. Finally, in section 3.3, we will introduce the different technologies that is used during the development of this system.

## 3.1 Choosing a Dataset

At the start of this master thesis, no dataset were yet available in the domain of palliative treatment of lung cancer patients. We therefore had to find another suitable dataset in order to test our integration of BN and CBR. We searched for datasets dealing with cancer and palliative care at the UCI Machine Learning Repository [53]. We found that most of the datasets relating to diseases were very complex and dealing with unfamiliar domains, and it would therefore be difficult to create and quality assure cases and a Bayesian network.

Instead, we decided to use the Wine Quality datasets because it appeared less complex plus having a more familiar domain. The Wine Quality dataset actually consists of two datasets: one for red wine and one for white wine. The red wine dataset has 1600 instances, while the white wine dataset has almost 4900 instances. Both datasets have 12 feature-value attributes, 11 of these are input variables (representing acidity, sulphates etc.) and the 12th represents the quality score assigned by a panel of wine experts. These datasets have resemblance to palliative treatment since they have defined outcomes (i.e. the quality) for the instances in the same way as a palliative treatment will have (i.e. the degree of pain). This resemblance will hopefully also make it easier to adapt the system to the palliative dataset. Another favorable property is the large amount of papers that have been published in the area of wine quality. This makes it substantially easier to extend the data when necessary.

One benefit of this dataset is that they are represented in the csv file format, which is supported by GeNIe (i.e., one of the framework that should be used in the development of the system). By using one of these datasets, we can insert it into GeNIe and generate probabilistic models to be used in the system. This way, we can both generate cases and create the basis for the Bayesian Network with use of these dataset.

Although, when working with the dataset, we decided that we not wanted to create the BN out of the case base. Instead of having the same attributes in the BN and the cases, we want the BN to represent general information about cause and effect, while the case represents a specific instance. This became difficult using this dataset (or any other dataset from UCI). We therefore decided to create our own model based on our own knowledge and interest. Tore Bruland, one of the co-supervisors, suggested that we could combine the wine domain with the meal domain. The topic of finding an appropriate red wine to a given meal was therefore chosen. This topic could be understood by most humans with a slight interest in red wine and has a vast amount of knowledge available on the Internet and in books.

We wanted to build a system that utilizes both the known dependencies between food and wine that can be represented by a BN, and a number of recipes with matching wines in the case base. We therefore combined two datasets: one that represents the meal (i.e. the case base) and one that represents the grape and its characteristics (i.e. the Bayesian network).

The combination of CBR and BN can give an added value to this domain. While CBR can give us specific cases, the BN can give us general directions about how to pick an adequate wine. A pure CBR system would only be able to return specific cases matching the given input values.

## 3.2 Fundamental Design

Our system is called Bacchus after the Roman god of wine. An overview of the system is illustrated in figure 3.1. This design is similar to the architecture we proposed in the previous specialization project (see section 2.3), except that is has been slightly transformed to suit the topic of finding a adequate wine for a given meal. The most obvious difference is the omission of the clinical guidelines. These guidelines were independent of the CBR and BN part of the system, and can therefore easily be replaced or removed. Bacchus could also have included guidelines representing general rules of combining food and wine, but we chose to focus on the basic parts of the system. As our task is to implement an integration of CBR and BN, we wanted to evaluate it on its own before adding more methodologies. This expansion is therefore left as further work.

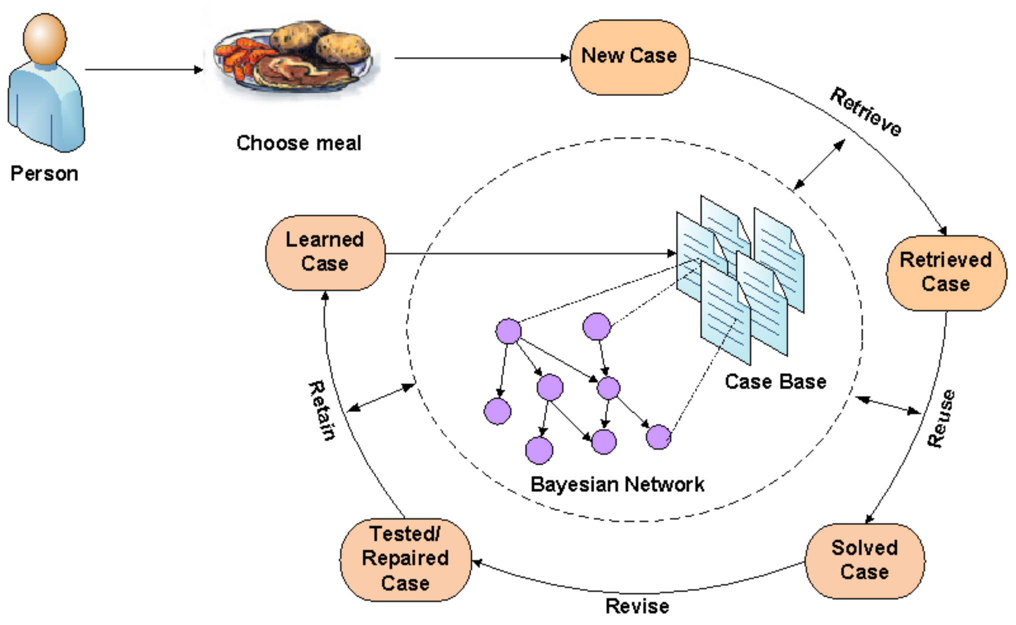


Figure 3.1: The fundamental design of Bacchus.

Bacchus thus remains with the core technology from our clinical decision support architecture, namely CBR and BN. In order to chose an adequate wine for a meal, you need to have some background knowledge about which characteristics the wine should have to suit the characteristics of the meal. For example, should food with herbal character be accompanied with wine with herbal character (i.e., red wines from South Africa). It is also important to balance the relations between the fat contents of the meal and tanning agents in the wine (i.e., a meal with high fat contents requires a wine with rich tanning agents). Two apparently different meals could therefore have the same characteristics and the same wine recommendations. This way, the BN can have a bridge-builder function between cases in the case base.

The system starts with a person choosing a meal. This meal will constitute a new case in the case base. Features of this case will be used to retrieve other cases. During the retrieve phase, the BN will be used to help guide the selection of cases. The BN will contain general information about combinations of food and meals which the system can utilize to find new solutions. This gives the system the ability to draw parallels between cases on the basis of similarities in the BN. We believe that this could significantly improve the results of this system. After the system has retrieved the most relevant case(s), a solution can be refined by using the BN to justify the solution(s) and picking the most rational explanation(s). The explanation(s) will hence influence the adaption of the solution case. The revision phase is normally done by an expert, and is omitted in many system for that reason. This step can be executed in many different ways, depending on the actual use of the system. Two possible solutions could be to either use a wine expert (e.g., an employee at Vinmonopolet), or let the user revise the solution himself. The tested/repaired case should then be retained by the system in order for the system to learn and develop. The case is saved in the case base and the Bayesian network is updated if necessary based on the information about the meal and the wine in the case.

## 3.3 Technologies used

This section will introduce the different frameworks used in the development of our system. We were recommended to use jColibri and GeNIe & SMILE by our supervisors. In addition to these, we had to do some technical choices regarding the representation of the cases and the choice of similarity measures.

There are several different ways to represent cases in a CBR system. jColibri supports three types of representations: textual cases, cases stores in a database and cases stored into ontologies. We chose to store the case base in a database in order to support a large case base with possible extensions in the future.

### 3.3.1 jColibri

jColibri is a framework for building CBR systems. jColibri grew out of the Group for Artificial Intelligence Applications (GAIA) at the Complutense University of Madrid. The GAIA group consist of a group of professors and graduate students that aims to advance the state of art in AI research with focus on case-based reasoning, knowledge acquisition and machine learning. The framework integrates well proven software engineering techniques (e.g., object orientation), making it easy to reuse and extend the system. jColibri is designed with a clear separation of the problem solving method and the domain model, which is the Knowledge Acquisition and Documentation Structuring (KADS) key idea to flexible use and reusability of domain knowledge.

One of the main features of jColibri is that it is oriented both to developers and designers. This is done by its two layered architecture where the bottom layer provides an architecture to build CBR applications, while the top layer contains composition tools to generate CBR applications. Our work will therefore be concentrated in the bottom layer. This layer takes advantage of the possibilities in the newest Java 2 Enterprise Edition technologies, namely Hibernate and Java Beans. Hibernate is used to create the database connections and uses Java Beans to store the information in the database. By using these technologies, jColibri claims to support the development of commercial CBR applications [44].

There are several advantages by using jColibri to create the CBR part of our system. Firstly, jColibri supports ontologies, which is a representation of concepts and relationships between them. According to Recio-Garcia et al. [44], ontologies are useful in knowledge intensive CBR applications because they facilitate use of knowledge that is already acquired and thereby reducing the knowledge acquisition bottleneck (i.e., that nothing happens until the knowledge is maintained and processed). Ontologies are often used for similarity assessments in CBR applications. Ontologies can be created in Protégé (see section 3.3.3) and imported into jColibri.

jColibri has a subproject called OntoBridge that aims to ease the management of ontologies. OntoBridge lets us import ontologies to be used in our system. These ontologies can also be used to calculate similarity measurements. Built-in in jColibri are four concept based similarity functions: OntDeepBasic, OntDeep, OntCosine and OntDetail. These functions calculates the similarity based on the location of the cases in the ontology. The jColibri similarity functions is defined in Appendix A.

As mentioned, jColibri also supports the use of Hibernate, which simplifies the mapping between the database and the data model. jColibri handles persistency by using connectors. They are responsible to retrieve and return cases from the memory location to the CBR system. In addition to connectors used to retrieve textual cases, jColibri has a JDBC connector that makes it possible to use most of the available databases. Basically, jColibri aims to include different technologies and has therefore many choices to offer to the developer.

A pictured drawback with jColibri is that it can be complicated to get familiar with because of its size and its many functionalities.

The current version of jColibri is jColibri2. It is available for download at `http://gaia.fdi.ucm.es/grupo/projects/jcolibri/jcolibri2/index.html`.

### 3.3.2 GeNIe and SMILE

GeNIe (Graphical Network Interface) is a development environment that can be used to create graphical decision-analytic models (e.g., Bayesian networks). GeNIe is the graphical interface built on top of SMILE (Structural Modeling, Inference and Learning Engine), a Bayesian inference engine that can be used to build and edit graphical models and use them in reasoning and decision making. SMILE is implemented in C++, but has wrappers that makes the functionality available within application written in other languages such as .NET languages and Java [27].

GeNIe is able to learn probabilistic models from data, this way enhancing the ability to easily generate probabilistic models from large datasets. GeNIe and SMILE have been developed at the Decision Systems Laboratory at the University of Pittsburgh.

GeNIe is used to create the Bayesian network, while SMILE is used to do the necessary work on the model inside our system (i.e., retrieving probabilities etc.). A screen shot of GeNIe is illustrated in figure 3.2.

By using GeNIe, we are able to create a Bayesian model in the interface and test it before we import the model to our system. This interface also makes it easy to do changes on the model. When the model is working as expected, we can import it into our system and use SMILE to calculate probabilities and to reason with the model.

GeNIe and SMILE are available for download at `http://genie.sis.pitt.edu/`.

### 3.3.3 Protégé

Protégé is a Java-based open source software tool that can be used to develop knowledge-based systems and to create ontologies [17, 25]. Protégé is developed by Stanford Center for Biomedical Informatics Research at the Stanford University

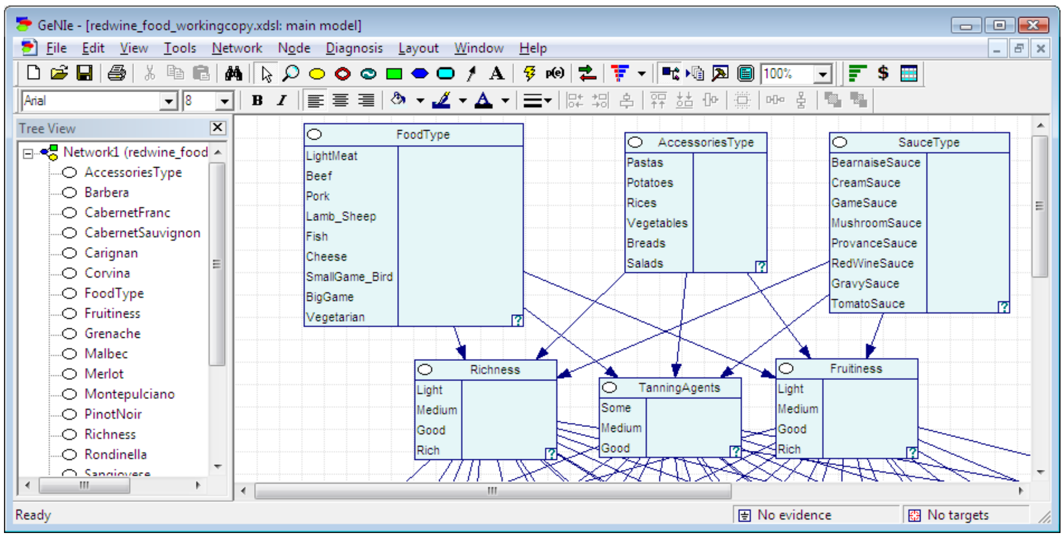


Figure 3.2: Screenshot of GeNIe

School of Medicine. The first version came out in 1987, and the current version is Protégé 4.1. It is available for download at http://protege.stanford.edu/.

By using ontologies, we can incorporate domain knowledge in our system in addition to the general knowledge in the BN. This could give the system an added value. The ontologies can also be used to calculate similarities with jColibri's built-in ontology similarity functions.

Figure 3.3 shows a screen shot of the OWL editor. This editor let us create ontologies in the W3C's Web Ontology Language (OWL). The ontology can contain description of classes, properties and instances. In the figure, we see the Food ontology in the Class browser to the left. This hierarchy illustrates the classes and the relationships between them. The main classes consists of the 9 main classes used by Vinmonopolet, and they are the same as the classes represented by the food node in the Bayesian network (see section 4.2). These main classes can also have sub classes, as LightMeat. Chicken, Turkey and Calf are all classified as LightMeat, and we separated these by subclasses (except Calf, which is an instance under LightMeat). Behind each class name, there is a number representing the number of instances in that class. To the right for the Class browser is the Instance browser. This browser shows the instances belonging in the selected class (Lamb_Sheep is the figure). To the left is the Individual editor where a instance can be edited.

In addition to be used to create ontologies, Protégé is also used together with myCBR to prototype the CBR part of the system.

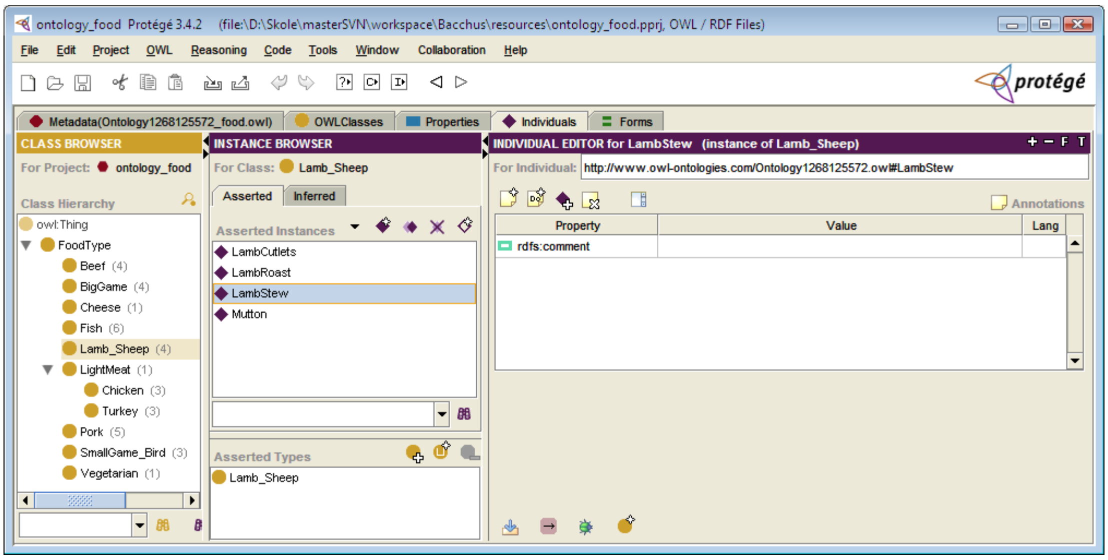


Figure 3.3: Screen shot of ontology editor in Protégé.

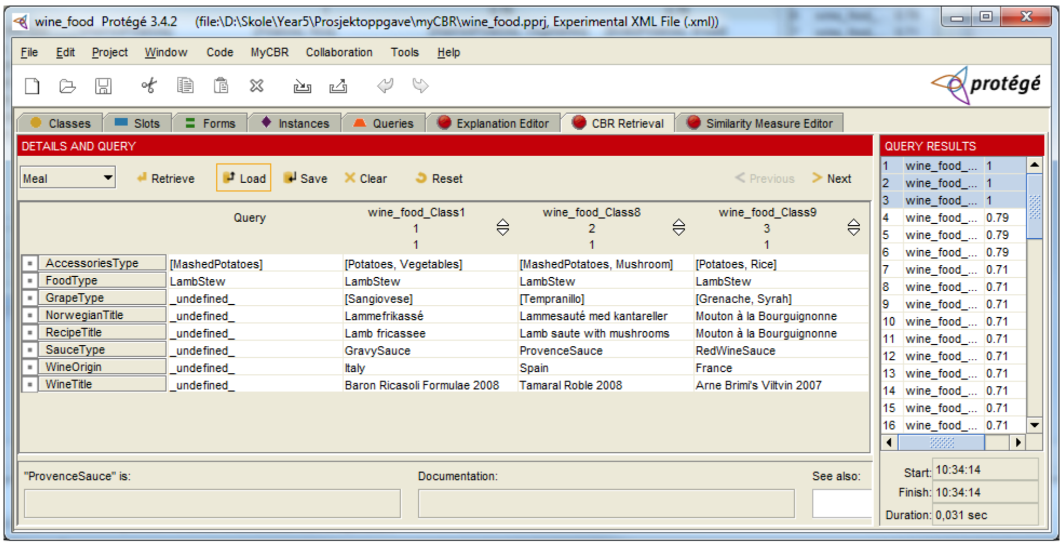
### 3.3.4 myCBR

myCBR is available as a plug-in to Protégé and can be used to create CBR systems. It is a CBR reasoning tool and it aims to enable fast prototyping, to be extendable and adaptable and to integrate state-of-the-art CBR functionality. myCBR is open source and is developed by the Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI) GmbH. It is available for download at `http://genie.sis.pitt.edu/`.

We will use myCBR to prototype the CBR part of our system and to create a similarity function to be used in the final system. By using myCBR to prototype, we can get feedback on what works and what that should be fixed. It also gives us the ability to compare our final system to a pure-CBR approach.

In figure 3.4, the tabs marked by red circular icons represents the myCBR part of the Protégé interface. These tabs handles the Explanation Editor, CBR Retrieval and the Similarity Measure Editor. myCBR focuses on the similarity-based retrieval phase, as this is the core functionality in many CBR applications [48].

The developers of jColibri and myCBR have together developed a wrapper that makes it possible to use similarity functions defined in myCBR in jColibri. This is a great advantage for us, making it possible to easily define our own similarity functions to be used in jColibri. Similarity functions can be exported in XML format and imported using the wrapper function.


wine_food Protégé 3.4.2 (file:\D:\Skole\Year5\Prosjektoppgave\myCBR\wine_food.pprj, Experimental XML File (.xml))
File Edit Project Window Code MyCBR Collaboration Tools Help
protégé
Classes Slots Forms Instances Queries Explanation Editor CBR Retrieval Similarity Measure Editor
DETAILS AND QUERY
Meal
Retrieve Load Save Clear Reset
Previous Next
QUERY RESULTS

| | Query | wine_food_Class1<br>1<br>1 | wine_food_Class8<br>2<br>1 | wine_food_Class9<br>3<br>1 |
|---|---|---|---|---|
| AccessoriesType | [MashedPotatoes] | [Potatoes, Vegetables] | [MashedPotatoes, Mushroom] | [Potatoes, Rice] |
| FoodType | LambStew | LambStew | LambStew | LambStew |
| GrapeType | _undefined_ | [Sangiovese] | [Tempranillo] | [Grenache, Syrah] |
| NorwegianTitle | _undefined_ | Lammefrikassé | Lammesauté med kantareller | Mouton à la Bourguignonne |
| RecipeTitle | _undefined_ | Lamb fricassee | Lamb saute with mushrooms | Mouton à la Bourguignonne |
| SauceType | _undefined_ | GravySauce | ProvenceSauce | RedWineSauce |
| WineOrigin | _undefined_ | Italy | Spain | France |
| WineTitle | _undefined_ | Baron Ricasoli Formulae 2008 | Tamaral Roble 2008 | Arne Brimi's Viltvin 2007 |

| | | |
|---|---|---|
| 1 | wine_food_... | 1 |
| 2 | wine_food_... | 1 |
| 3 | wine_food_... | 1 |
| 4 | wine_food_... | 0.79 |
| 5 | wine_food_... | 0.79 |
| 6 | wine_food_... | 0.79 |
| 7 | wine_food_... | 0.71 |
| 8 | wine_food_... | 0.71 |
| 9 | wine_food_... | 0.71 |
| 10 | wine_food_... | 0.71 |
| 11 | wine_food_... | 0.71 |
| 12 | wine_food_... | 0.71 |
| 13 | wine_food_... | 0.71 |
| 14 | wine_food_... | 0.71 |
| 15 | wine_food_... | 0.71 |
| 16 | wine_food_... | 0.71 |

"ProvenceSauce" is:
Documentation:
See also:

Start: 10:34:14
Finish: 10:34:14
Duration: 0,031 sec

Figure 3.4: Screen shot of the myCBR plug-in in Protégé.

### 3.3.5 Apache Derby

Apache Derby is an open source relational database developed in Java. Derby has an embedded JDBC driver, which provides support to embed the database in systems written in Java. Derby is available at `http://db.apache.org/derby/`.

Some of the advantages of Derby are: it is memory efficient, it is based on Java, JDBC and SQL standards, it offers embedded JDBC drivers (which makes it possible to integrate the database into any Java application) in addition to the traditional client/server mode and it is easy to install, deploy and use. As this project aims to implement and test a proposed integration of CBR and BN, we think that an embedded database will simplify the development process (contra having to set up a database server) while still giving the system the possibility to be easily up-scalable (by using a database instead of textual representation of cases).

### 3.3.6 Hibernate

Hibernate is an open source framework that simplifies the use of databases in Java by mapping the relational database to an object-oriented domain model. Hibernate can map Java classes to database tables, and also generate SQL queries automatically for the developer, thereby reducing much development time. In order to use Hibernate to manage the cases, the system must use Java Beans to represent cases. A Java Bean is a class that has a get() and set() method for each attribute.

By using Hibernate, we hope to reduce development time and simplify the work on the database. As our co-supervisor Tore Bruland also has experience with using Apache Derby together with Hibernate, we have access to support if needed. This also affected our selection of Apache Derby as our database.

Hibernate abstracts the choice of database, as it works with most databases. Other databases could be used in the system by simply changing the configuration files for Hibernate.

Hibernate is developed by Red Hat and can be downloaded from `http://www.hibernate.org/about.html`.

# Chapter 4

# Implementation

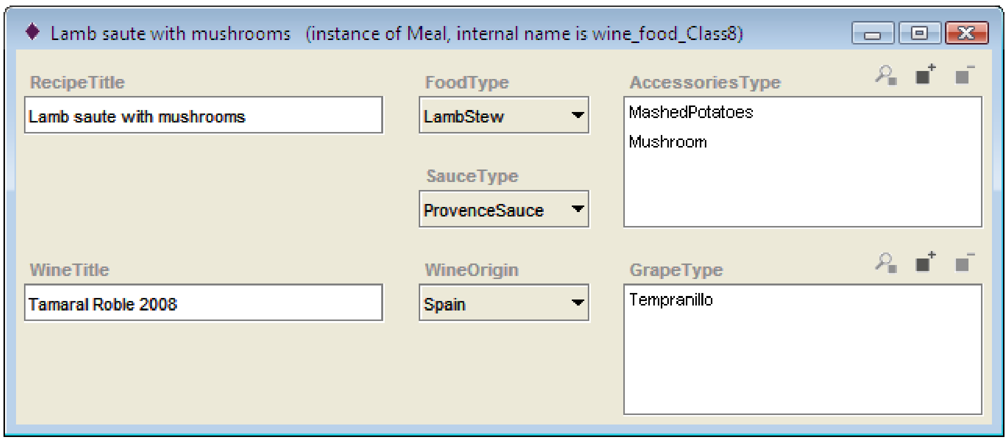
This chapter will describe the implementation of Bacchus. It is meant to give the reader insight into the design choices made under the development of the system. The first section will explain how the meal case base is created. Next, the development of the Bayesian network is described. In section 4.3, our system solution is thoroughly explained. Finally, in the last section, we will give the reader a guide through the system.

## 4.1 The Meal Case Base

The case base was created from Apéritif's[3] recipes available online. Apéritif is a large Norwegian web page about food and drinks and in addition to having a vast amount of recipes from different contributors, they also have wine suggestions for most of them. We chose 59 recipes that are divided among the 9 different food categories used in the BN, where each food type have between 5 and 8 instances. The recipes are chosen to create a versatile case base, and they all have belonging wine suggestions.

A case exists of a description of the recipe title, the meal items (i.e., the food type, the sauce and the accessories) and information about the wine (i.e., wine title, wine origin and grapes).

The case base was first modeled with myCBR. An example of a case instance in myCBR is illustrated in 4.1. The figure shows the 7 attributes of a case: the recipe title, the food type, the accessories types, the sauce type, the wine title, the wine origin and the grape types. Note that the case is an example from myCBR, but a case in the systems case base will have the same attributes. The recipe title and


Lamb saute with mushrooms (instance of Meal, internal name is wine_food_Class8)

RecipeTitle: Lamb saute with mushrooms
FoodType: LambStew
SauceType: ProvenceSauce
AccessoriesType: MashedPotatoes, Mushroom
WineTitle: Tamaral Roble 2008
WineOrigin: Spain
GrapeType: Tempranillo

Figure 4.1: A case instance. This is a screen shot from myCBR/Protégé.

the wine title are textual fields which describes the name of the recipe and the wine respectively. The rest of the fields are chosen from predetermined lists. Two attributes, food type and accessories type are represented by ontologies created in Protégé. We also chose to implement both accessories type and grape type as multi-valued attributes.

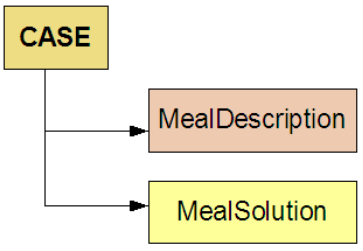
After creating the cases in myCBR, the case base was rewritten to SQL and inserted into our Apache Derby database (see section 3.3.5). The case base is then retrieved to the in-memory organization in jColibri by using Hibernate. Data structures are further elaborated in the next section.

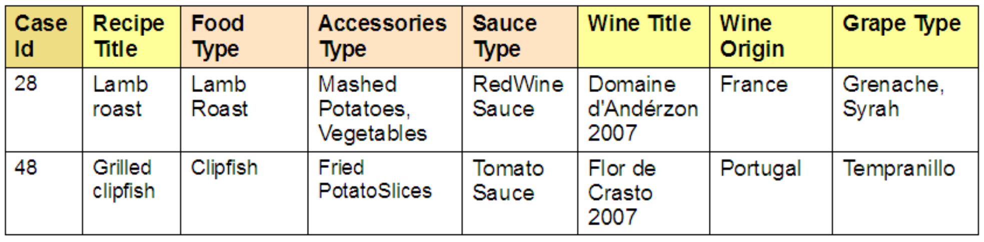
### 4.1.1 Data structures

The organization of the case base can affect the performance of the system. jColibri has four available in-memory organizations: a basic linear case base that stores the cases into a list, a cached linear case base that persists cases when closing the application, a ID indexed linear case base that uses the cases' ID to keep an index of the cases and a new clustered organization through the plug-in called Thunder [15]. We chose to use the linear case base, because we currently do not need a caching mechanism. The current size of the case base does neither require to index cases. Although, because of jColibris use of interfaces to define the properties of the case bases, we could easily switch to a different choice of case base when needed.

The case base have a flat, attribute-value representation. This is the simplest, straight forward case base organization. It is easy to store and retrieve, and thereby suitable for prototyping. As the current case base is designed, a flat case base is sufficient.

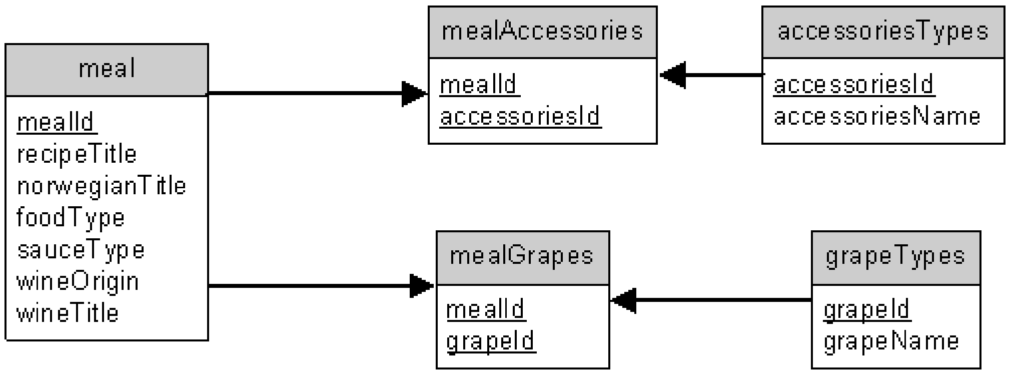
A case in jColibri contains four components: a description of the problem, a solution, a result of applying the solution and a justification of the proposed solution. Of these four, only the description is mandatory, so the rest could be left empty.





| Case Id | Recipe Title | Food Type | Accessories Type | Sauce Type | Wine Title | Wine Origin | Grape Type |
|---|---|---|---|---|---|---|---|
| 28 | Lamb roast | Lamb Roast | Mashed Potatoes, Vegetables | RedWine Sauce | Domaine d'Andérzon 2007 | France | Grenache, Syrah |
| 48 | Grilled clipfish | Clipfish | Fried PotatoSlices | Tomato Sauce | Flor de Crasto 2007 | Portugal | Tempranillo |

(a) Case structure



(b) Database tables

Figure 4.2: Data structures

Figure 4.2 shows the structure of the cases and the database. As shown in figure 4.2a, our cases exist of a description and a solution. The description consists of the food item, the accessory and the sauce, and is used to retrieve similar cases. The solution consists of information about the accompanying wine (i.e., wine title, origin and grape types) and the recipe title.

As mentioned, a case in jColibri also has a justification of the solution and the result of applying the solution. Kim Pedersen has been working on generating justifications (i.e., explanations) of the solutions created by our system. The resulting component can be used to reveal if the solution was successful or not. This part is not implemented in our system.

In figure 4.2b, we can see the corresponding data base tables. The main table where the meal is represented is `meal`. In order to support multiple accessories and grapes per meal, we separated these attributes into own tables. `mealId` in the data base tables corresponds to `caseId` in the cases. The attributes in the database have corresponding names in the case structure.

## 4.2 The Meal BN

We wanted to build a Bayesian model that represents the connections between the food and the characteristics of the wine. In order to create the model we studied the dependencies in the wine domain. Several different ways of measuring the wines characteristics exists.

Vinmonopolet has the exclusive right to sell wine in Norway, and are own by the state. In addition to sell wine and other alcoholic beverages, Vinmonopolet also practices a lot of educational work. They publish their own magazine, Vinbladet, with six editions per year, and there are also a lot of information available at their web pages. In order to give our model an reliable source, we used this information in the development of the Bayesian network. We chose to use Vinmonopolets [54] classification of *richness*, *tanning agents* and *fruitiness* in our model because of its simplicity. This also gives us the advantage that we could use Vinmonopolets supplementary information available both on-line and in their published magazines. The characteristics of the wine does not only depend on the food type, but also on what accessories and sauce that it is served with. The characteristics will further influence the choice of grape types, seeing that the different grapes often have special characteristics. The output of the BN will be the probabilities of the grapes matching the meal ingredients. The input to the BN will therefore be the same as a MealDescription in figure 4.2a (i.e., food type, accessories type and sauce type). The BN is illustrated in figure 4.3. The non-observable middle nodes in the figure are not represented in the cases as they represent most adequate grape characteristics given the meal contents. The output nodes, the grapes, are found in a case. In the BN, they represent proposed solutions to a meal. In a case, grapes are also part of the solution, but here as the grapes present in the proposed wine (e.g., the Gran Coronas Reserva 2005 is goes to the meal Stewed beef and contains the grapes Tempranillo and Cabernet Sauvignon).

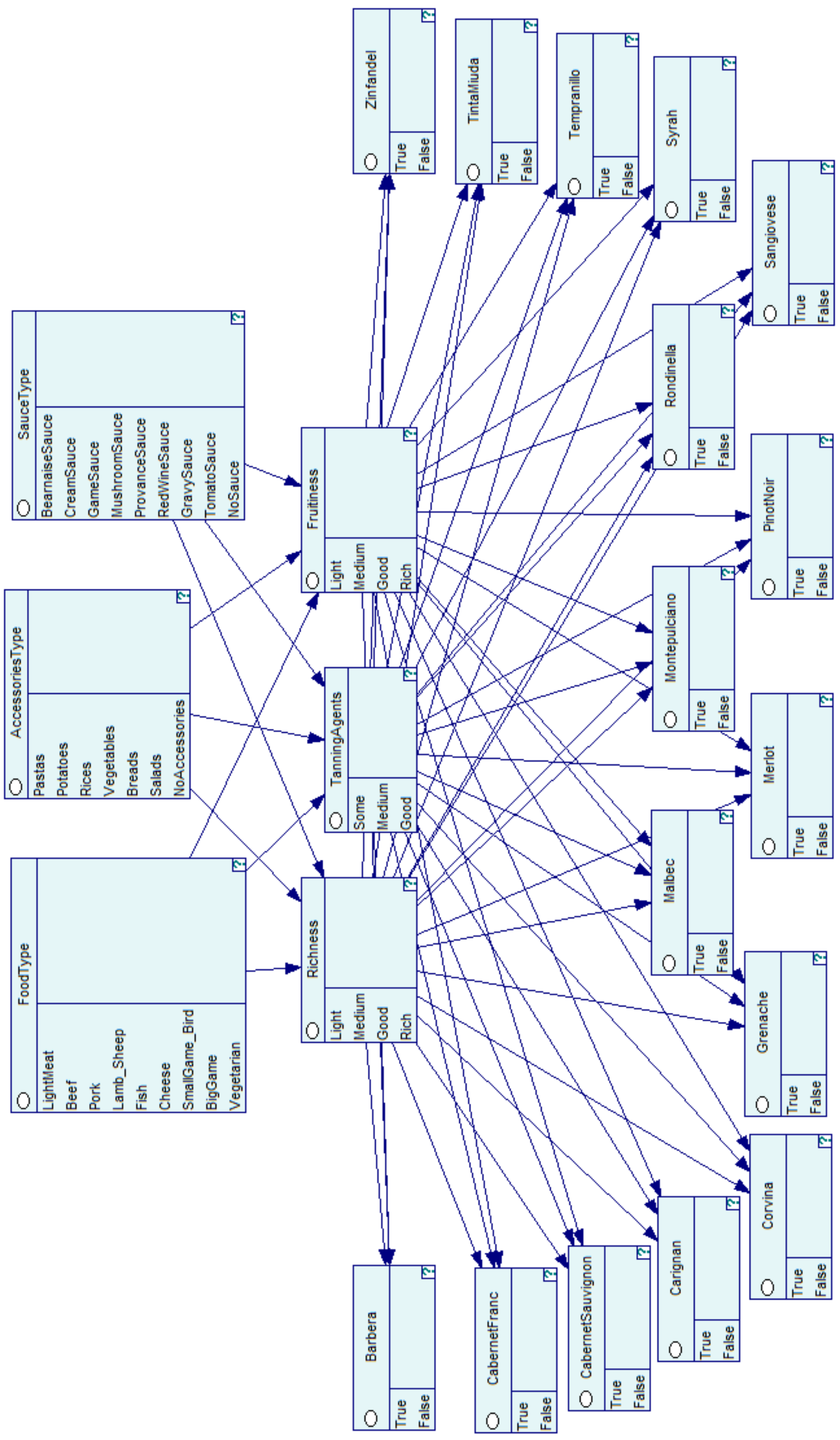


Figure 4.3: The first design of the BN

We used information from Vinmonopolet [54] to fill in the probabilities in the BN. Vinmonopolet has a detailed overview of all their wines available online with information of what food a specific wine goes to, its grapes and its characteristics. In order to get the probabilities for the grapes, we used the search available online and listed all available wines with a specific grape. Then, we counted the instances of each specific characteristic. The model is based on the wine assortment found at Vinmonopolet in March 2010. Because of constant adjustments in their assortment, this model will probably not be 100 % consistent in the future. The probabilities will still basically be the same, as they represent the general characteristics in the different grapes.

The numbers were normalized in order to get even probabilities across grapes with unequal occurrences. For example, if we choose *richness*: `good`, *tanning agents*: `good` and *fruitiness*: `rich`, the Barbera grape gets 24.5% probability, while Sangiovese gets 2.9%. These probabilities means that if you choose a random Barbera wine, you will have a 24.5% chance of choosing a wine with these characteristics.

The reader should note that the search resulted in all bottles with the specific grape, independent of the amount of that grape. This means that a bottle that contains 5 % of that grape will be on a par with bottles containing 80 % of the same grape.

The function we used to calculate the probabilities was:

$$P(grape|characteristic) = \frac{\text{\# of bottles with characteristic for this grape}}{\text{total \# of bottles with grape}} \tag{4.1}$$

where *characteristic* is the given *richness*, *tanningAgents* and *fruitiness* and *# of bottles with characteristic* is the number of bottles of the given *grape* and with the specific *characteristic* and *total # of bottles with grape* is the total number of bottles for the *grape*.

For Barbera, this would be:

$$P(Barbera|richness = good, tanningAgents = good, fruitiness = rich) = \frac{\text{45 bottles with this characteristic}}{\text{184 bottles total for Barbera}} = 24.5\% \tag{4.2}$$

Information from Vinmonopolet was also used to fill out the relationships between food types and characteristics. The first design of the model had direct connections between food, accessories and sauce and richness, tanning agents and fruitiness (see figure 4.3). These numbers were found by choosing a food type (i.e. light meat, fish etc.) in their assortment section on the Internet and observing the distribution between the characteristics. The following function were used to calculate the probability of a characteristic:

$$P(characteristic|food) = \frac{\text{\# of bottles with characteristic}}{\text{total \# of bottles adequate for this food}} \tag{4.3}$$

where *characteristic* is the given *g.richness*, *g.tanningAgents* and *g.fruitiness* and *# of bottles with characteristic* is the number of bottles that goes to the *food* and with the specific *characteristic* and *total # of bottles adequate for this food* is the total number of bottles adequate for this *food*.

In this model, we had some problems with quantifying the influence from sauce and accessories on grape characteristics (i.e. richness, tanning agents and fruitiness). Kim Pedersen (our collaborating student) therefore wanted to extended this model with three extra nodes, representing the bitterness, freshness and the flavor of the accessories and the sauce. By using these nodes, we were able to classify these characteristics of the sauces and the accessories in a broader manner and thereby simplifying the complex probability tables for *richness*, *tanning agents* and *fruitiness*. Pedersen also reduced the degree of dependency between the nodes. In the first design, each of the grape characteristic nodes were dependent of all three food items. In the new design, *richness* only depends on food type and flavor, *tanning agents* depends on food type, bitterness and flavor and *fruitiness* depends on food type and freshness. The new design is illustrated in figure 4.4, where the new nodes are marked by a red ring surrounding them.

## 4.3 System solution

The architecture is based on the findings in Gravem [18]. As explained in 3.1, we could not get the medical data in time, and we therefore chose to use the domain of finding an adequate red wine to a meal. This architecture was adapted to this domain. An illustration of the conceptual architecture is given in figure 4.5.

Our design use jColibri and Smile to do the reasoning, and we have separated them by using different background colors in the figure. The jColibri part has been divided into squares to separate the different sections in the system. A CBR application created in jColibri must contain the four parts (i.e. methods): CONFIGURE, PRECYCLE, CYCLE and POSTCYCLE. In our figure, we chose to omit the POSTCYCLE part since it only closes the system. Instead, we chose to represent QUERY in order to better illustrate the basic information flow in the system. In the figure, information flow is represented by dotted arrows (i.e., the arrows from Hibernate and Apache Derby to the case base in PRECYCLE

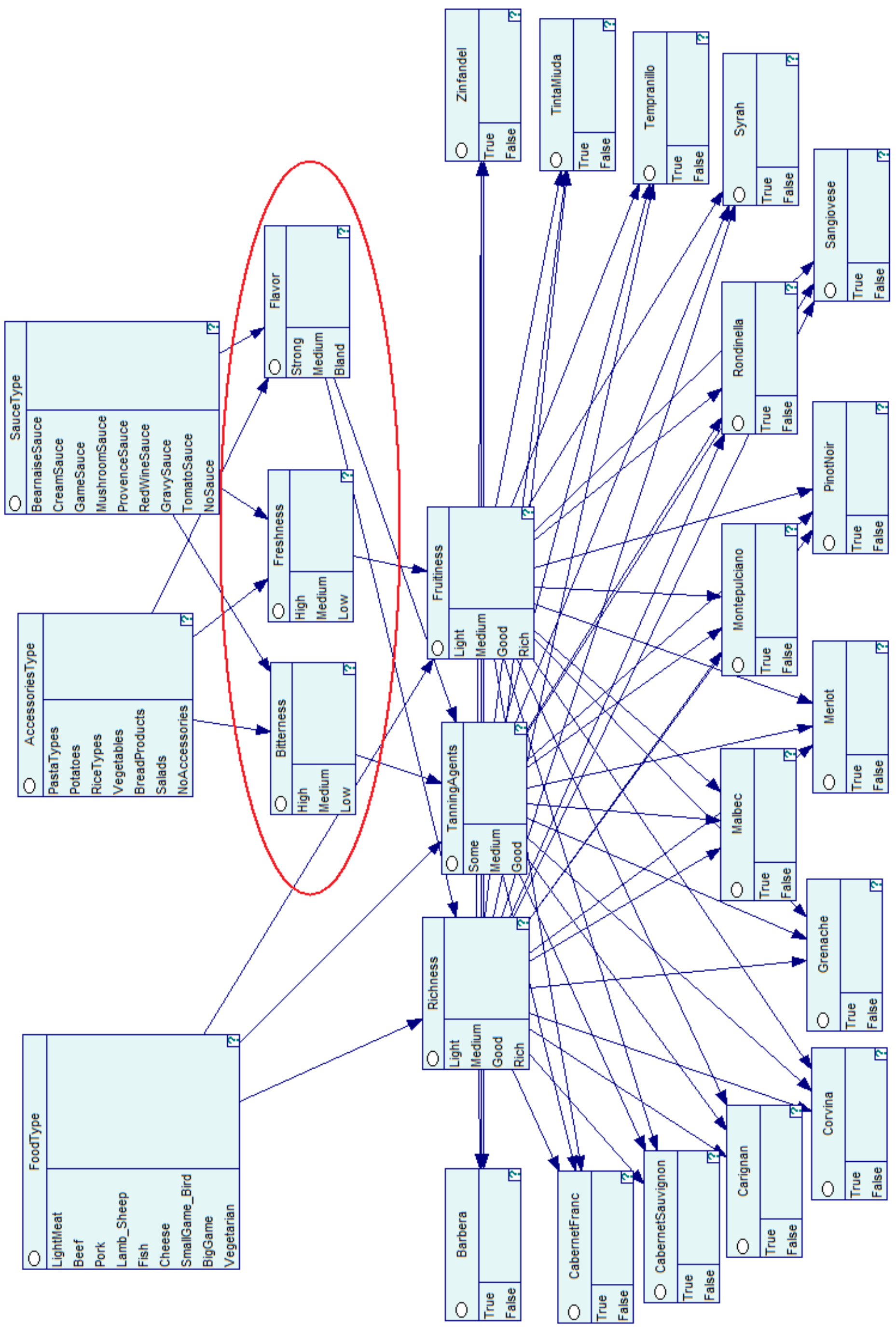


Figure 4.4: The Bayesian model after the extension.

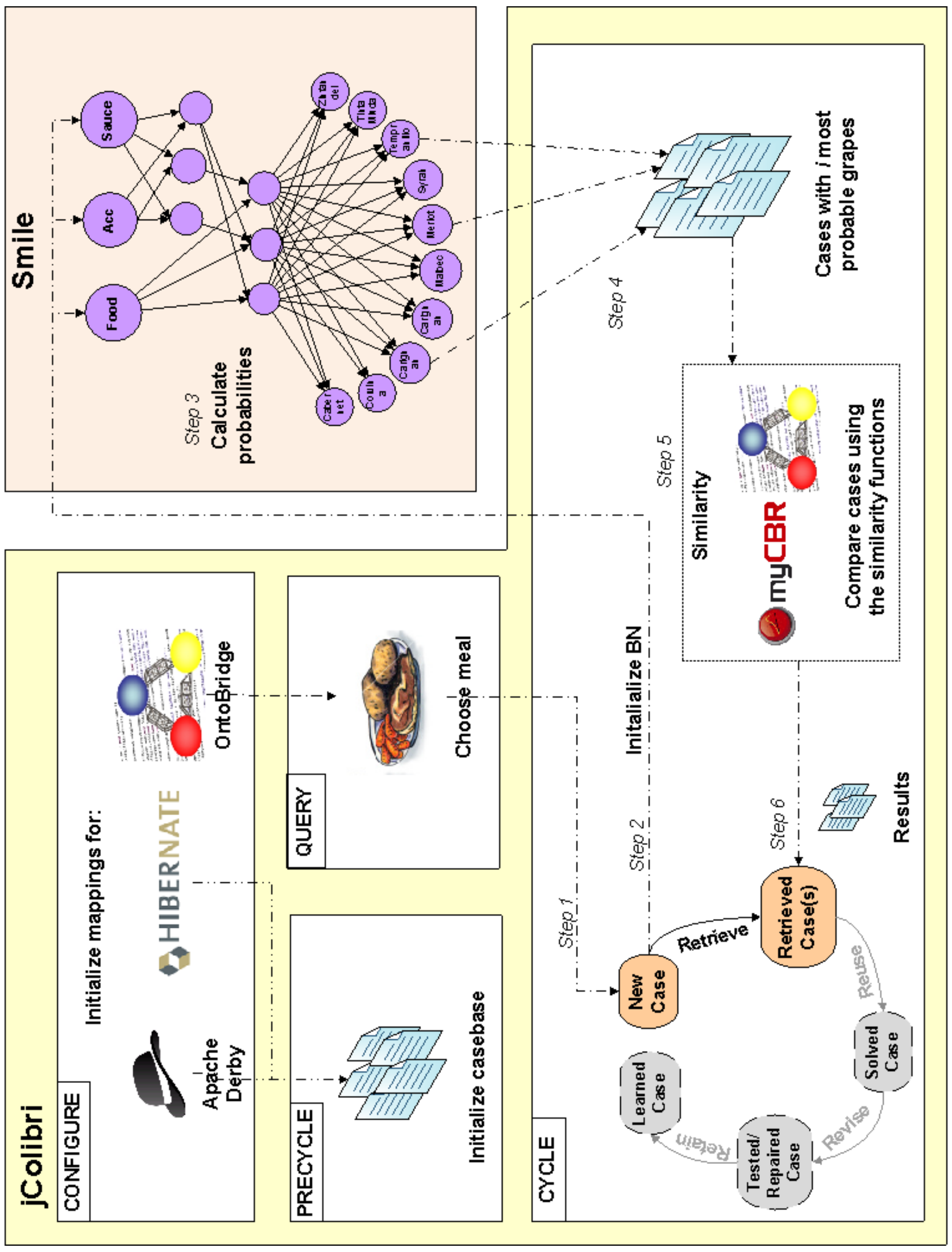


Figure 4.5: The conceptual architecture of Bacchus.

representing that the case base uses Hibernate to import the case base from the Apache Derby database). We have grayed the parts of the CBR cycle not yet implemented (i.e., reuse, revise and retain).

jColibri is the basis of our system and it starts with the CONFIGURE part. After this is done, the PRECYCLE part is executed. Then comes the QUERY and afterwards the main phase, CYCLE. These parts will be described more thoroughly in sections 4.3.1 to 4.3.3.

### 4.3.1 Configure

This phase does the configuration of the application. First, a connection to the Apache Derby Data Base is established and the SQL script is run, creating the tables and inserting the cases. After the database is set up, the database connector is initialized. This connector manages the persistence of the cases by accessing and retrieving cases in a uniform way. This connector uses Hibernate. The database connector is initialized from a XML file and maps the database tables to the corresponding classes by using Hibernate.

In the last part of the configure-phase, OntoBridge is set up. OntoBridge is a library that manages ontologies and is a subproject of jColibri. Our system uses ontologies to define the food and accessories types. These two ontologies were chosen because the instances can be put into main groups (i.e. Mashed potatoes belong to the group Potatoes). In addition, we could also defined sauces using ontologies by separating named sauces by their sauce base (i.e. white, brown etc). This would have increased the complexity of the system, so we chose to only represent the basic sauces. The ontologies for the food and accessories are first created in Protégé [25] and then loaded into our system. The database connector is then able to link the values in the database with the instances in the ontology by using OntoBridge. The ontologies are used both to pick a food and an accessories item, and to calculate the similarity of the cases.

### 4.3.2 Precycle and Query

The PRECYCLE phase loads the case base from the database and into memory. As the dotted arrows pointing to PRECYCLE in figure 4.5 shows, this phase uses the database connector created in the CONFIGURE phase (represented by the Hibernate logo in the figure) to read the cases from the database. Cases are then represented as MealDescription and MealSolution objects, as illustrated in figure 4.2a.

The QUERY is where the meal ingredients (i.e. food, accessories and sauce) is chosen. The ontologies are used to pick the food and accessories that the meal should contain, while the sauce is picked from a list of basic sauces. For simplicity, we only implemented the ability to chose one food item or one accessories item. The meal ingredients are sent to the CYCLE phase and constitute the basics of a new case.

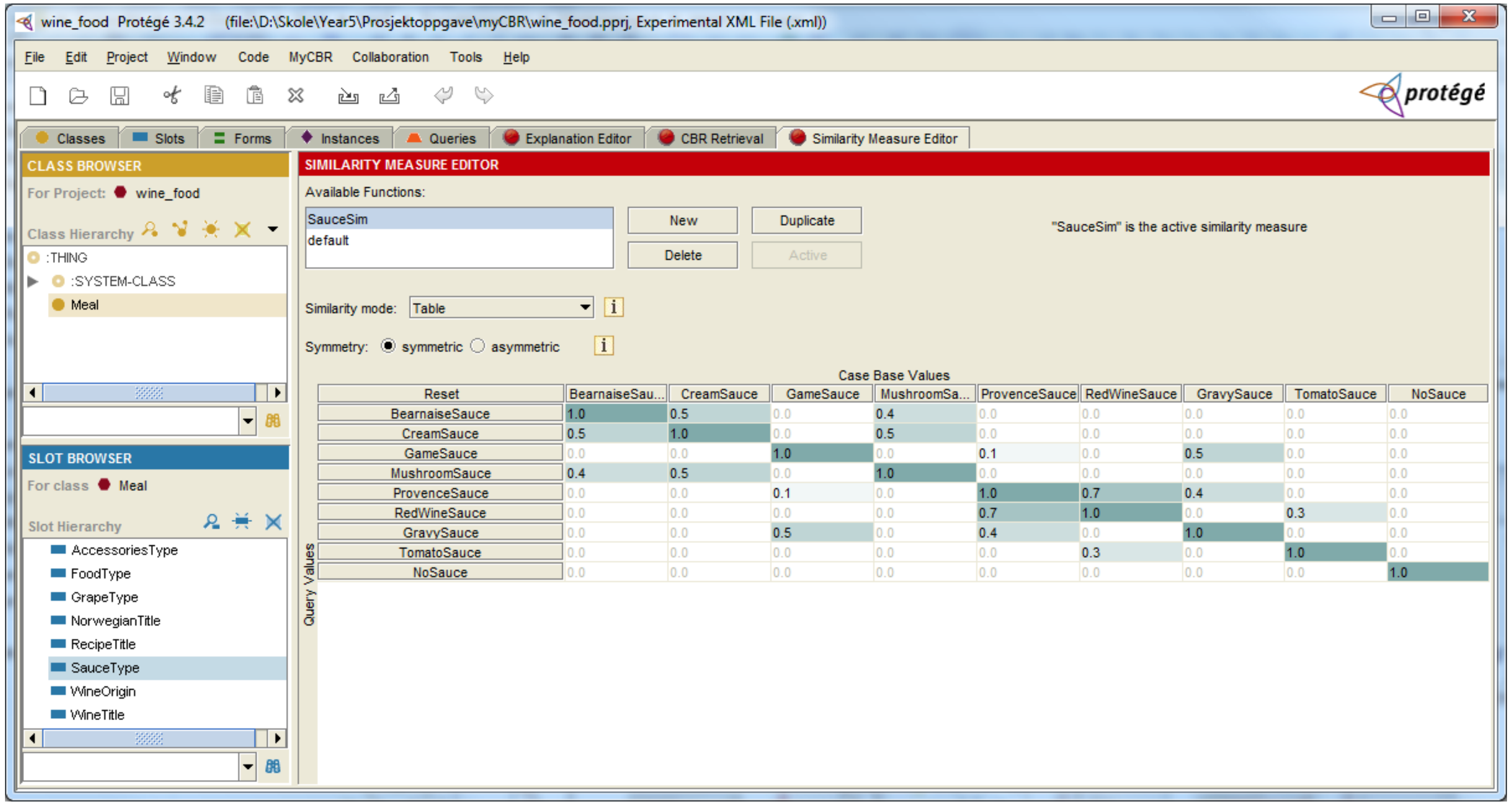
### 4.3.3 Cycle

This phase executes the CBR cycle. This part of the figure is marked with steps (1-6) for clarity.

In *step 1*, the meal ingredients chosen in QUERY is used to form a new case description (i.e., a MealDescription). Next (*step 2*), the SMILE framework is used to initialize the Bayesian Network representing dependencies between the different meal items and the wine grapes. This BN has been created in advance with GeNIe and the network-file is used to initiate the meal network. In *step 3*, the meal ingredients chosen is set as evidence in the meal network and SMILE calculates the probabilities for each grape. The cases that contains at least one of the $i$ most probable grapes are picked out for further reasoning (*step 4*), where $i$ is a number that is chosen by the executor.

These cases are compared using similarity functions in jColibri (*step 5*). jColibri has both local similarity functions that compute the similarity for each attribute and a global similarity function that computes the similarity for compound attributes. As local similarity functions, we defined three different types: ontologies, myCBR similarity and object comparison. jColibri has four functions that computes the similarity of cases based on their location in the ontology: OntDeepBasic, OntDeep, OntCosine and OntDetail. These functions can be used to compute similarity for the attributes that are defined by ontologies. We chose to use all four of these ontology similarity functions, and leave the evaluation of the functions to the user. These ontology functions do only support cases with single-valued attributes. We therefore had to extend these classes in jColibri to be able to deal with multi-valued attributes (i.e., accessories in a meal). jColibri also has a similarity function named Equal that compares objects. The jColibri similarity functions we used can be found in appendix A.

In addition, we defined a similarity function in myCBR. As mentioned, the people behind jColibri and myCBR have developed a wrapper that makes it possible to use myCBR similarity functions with jColibri. We used myCBR to create a table similarity function for sauce type, which is the only meal description attribute that


wine_food Protégé 3.4.2 (file:\D:\Skole\Year5\Prosjektoppgave\myCBR\wine_food.pprj, Experimental XML File (.xml))
File Edit Project Window Code MyCBR Collaboration Tools Help
protégé
Classes Slots Forms Instances Queries Explanation Editor CBR Retrieval Similarity Measure Editor
CLASS BROWSER
For Project: wine_food
Class Hierarchy
:THING
:SYSTEM-CLASS
Meal
SLOT BROWSER
For class Meal
Slot Hierarchy
AccessoriesType
FoodType
GrapeType
NorwegianTitle
RecipeTitle
SauceType
WineOrigin
WineTitle
SIMILARITY MEASURE EDITOR
Available Functions:
SauceSim
default
New
Duplicate
Delete
Active

"SauceSim" is the active similarity measure

Similarity mode: Table

Symmetry: [✓ symmetric] asymmetric

Case Base Values

Query Values

| Reset | BearnaiseSau... | CreamSauce | GameSauce | MushroomSa... | ProvenceSauce | RedWineSauce | GravySauce | TomatoSauce | NoSauce |
|---|---|---|---|---|---|---|---|---|---|
| BearnaiseSauce | 1.0 | 0.5 | 0.0 | 0.4 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| CreamSauce | 0.5 | 1.0 | 0.0 | 0.5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| GameSauce | 0.0 | 0.0 | 1.0 | 0.0 | 0.1 | 0.0 | 0.5 | 0.0 | 0.0 |
| MushroomSauce | 0.4 | 0.5 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| ProvenceSauce | 0.0 | 0.0 | 0.1 | 0.0 | 1.0 | 0.7 | 0.4 | 0.0 | 0.0 |
| RedWineSauce | 0.0 | 0.0 | 0.0 | 0.0 | 0.7 | 1.0 | 0.0 | 0.3 | 0.0 |
| GravySauce | 0.0 | 0.0 | 0.5 | 0.0 | 0.4 | 0.0 | 1.0 | 0.0 | 0.0 |
| TomatoSauce | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.3 | 0.0 | 1.0 | 0.0 |
| NoSauce | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 |

Figure 4.6: The similarity measure for the sauce types.

is not represented by an ontology. jColibri has the built-in function Equal, that compares two strings. This will not be able to represent the relations between the different sauces and we therefore decided to create our own function. A screen shot of this similarity function is found in figure 4.6. We used the table similarity function because the similarity is based on human knowledge (i.e., the red wine sauce and the Provence sauce is very similar to each other) and the relationships between the different sauces is not easily represented by mathematical functions. As we can see from the figure, we created a symmetric function, meaning that the red wine sauce has the same similarity to Provence sauce as the Provence sauce has to red wine sauce (i.e., 0.7). The similarity function is exported as a XML-file to our system, where the wrapper is used to calculate the similarities.

We chose to use the average function as our global similarity function, meaning that it will calculate the average values of all the individual similarities. Each individual similarity consists of a calculated local similarity and a weight that represent the importance of the specific entity. This weight can be used to set different importance on food, sauce and accessories. The result from the global similarity function will be used to differentiate between the cases.

In *step 6*, the top *k* cases are retrieved for further reasoning based on the similarity calculations in *step 5*. As we can see from the figure, retrieve is currently the only implemented step of the CBR cycle.

## 4.4 Running Bacchus

This section will guide the reader through the use of the system. The implemented system is attached to this report by a CD. The system can be run through the jar file `bacchus.jar`. This jar-file contains all that is needed to test the system. The source code is also attached in the folder `Source Code` on the CD. A reader interested in investigating the system or developing it further can import the workspace into an software development environment such as Eclipse.

We created a GUI for the system in order to create a more user-friendly application. We chose to use the same GUI as jColibri's TravelRecommender in order to reduce the amount of work. This GUI was therefore adapted to suit our system.

Relevant attributes from the illustrations are marked by using `typewriter` text. This is done for clarity for the reader. All dialogs have an `Exit` button that can be used to close the program.

In this review, we want to find an adequate wine for a lamb stew with mashed potatoes and Provence sauce.

### 4.4.1 Choosing a meal

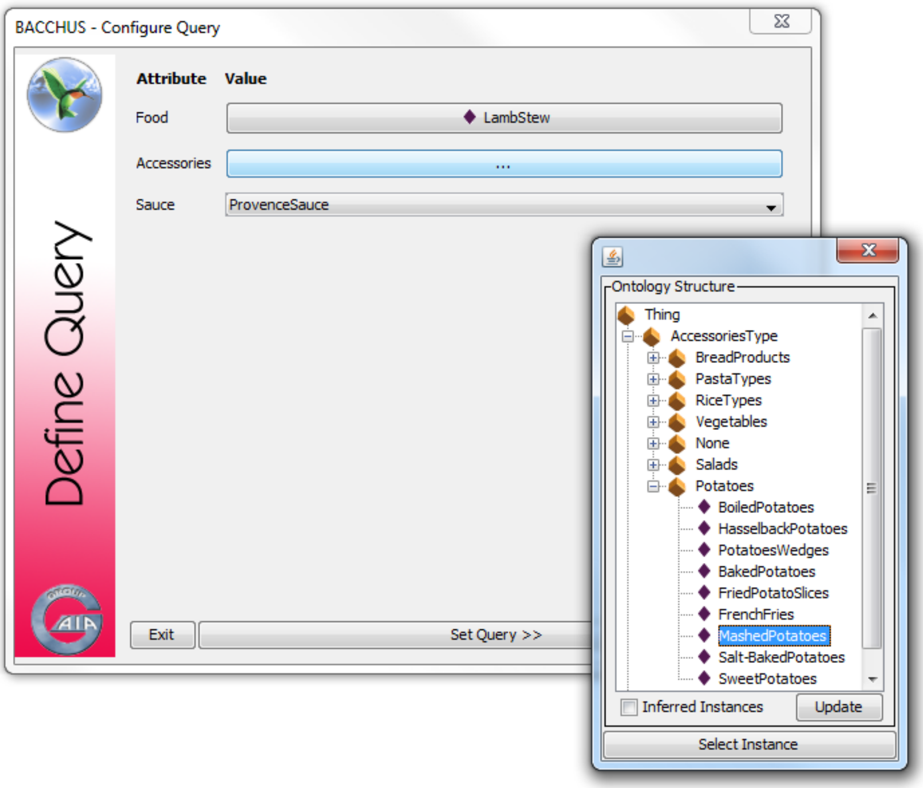


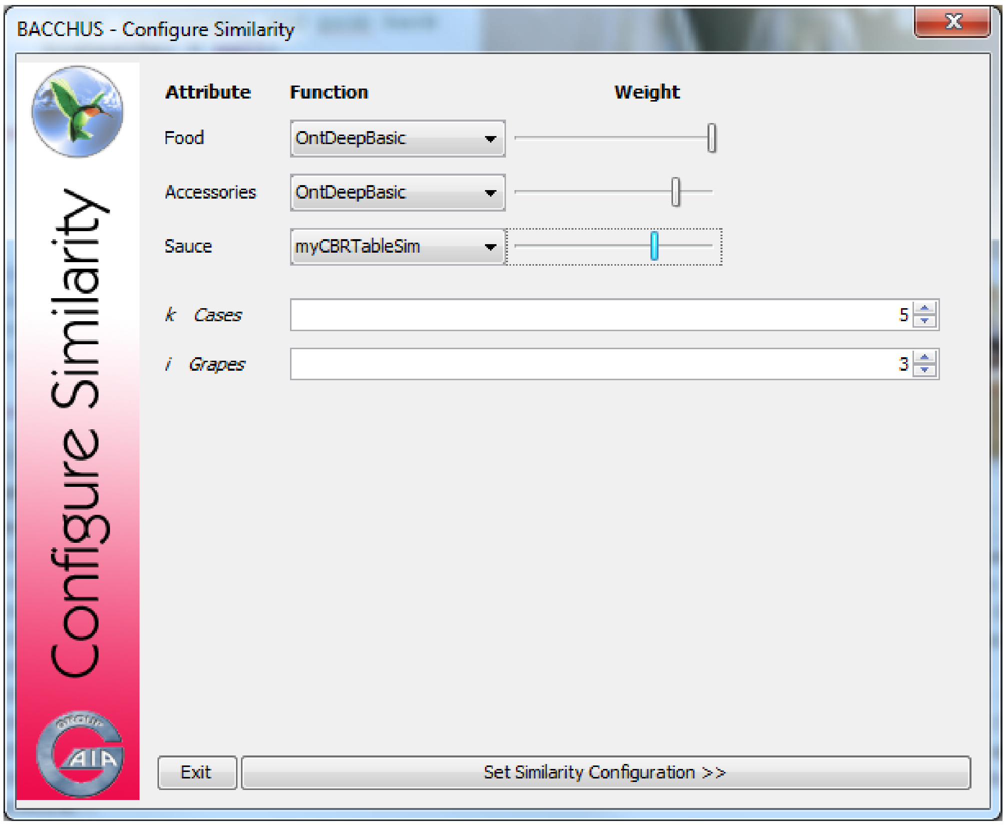
Figure 4.7: The Query Dialog.

The first dialog is where the user selects the contents of the meal. This dialog is illustrated in figure 4.7. `Food` and `Accessories` are both connected to the ontologies (explained in section 3.3.3). As we can see from the figure, the Ontology dialog pops up when the user pushes the `Accessories` button (or the `Food` button). This dialog shows the concepts and the instances in the ontology. Only instances marked with a purple diamond can be chosen. `Sauce` is a combo box and contains the same sauces as defined in the Bayesian network.

The user can choose if he/she want to specify the food and the accessories. The sauce is currently required because of the myCBR similarity function, and will result in a myCBR similarity function error if not defined.

After the meal items are chosen, the user presses the `Set Query>>` button to go to the next dialog.

### 4.4.2 Defining the similarity


BACCHUS - Configure Similarity

Configure Similarity

| Attribute | Function | Weight |
|---|---|---|
| Food | OntDeepBasic | |
| Accessories | OntDeepBasic | |
| Sauce | myCBRTableSim | |

*k Cases* 5

*i Grapes* 3

Figure 4.8: The Similarity Dialog.

The similarity dialog (see figure 4.8) is used to defined the similarity. Each of the attributes from the *Query Dialog* has an associated function and a weight. As mentioned in 4.3.3, we have six different similarity functions. `Food` and `Accessories` are ontologies, and they have therefore special ontology similarity functions (i.e., OntDeepBasic, OntDeep, OntCosine and OntDetail). In addition, they have associated a function that compares objects: Equal. `Sauce` is not an ontology, so it has no ontology functions associated with it. Instead, it has the same Equal function, and also the myCBR similarity function that we created: myCBRTableSim (see figure 4.6). As mentioned, each of the similarity functions also has a weight. These weights are used to set the importance of the different attributes in the calculation of the case similarity. It is natural to dedicate the highest weight to `Food`, since it has the greatest influence on the choice of wine. `Accessories` and `Sauce` should then have lower weights. We also define the $k$ number of cases that should be retrieved and the $i$ number of grapes that should be used in the further reasoning.

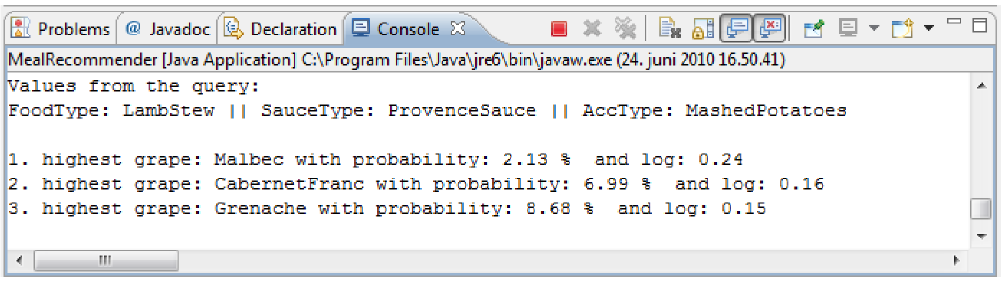
This dialog is predefined with OntDeepBasic as similarity function for `Food` and `Accessories`, and myCBRTableSim for `Sauce`. All similarity functions are predefined with weights being 100 %. The $k$ number of cases are predefined to 5, while the $i$ number of grapes are predefined to 3. These settings can be changed by the user, but can also be used as they are. We particulary recommend the user to adjust the similarity weights.

In figure 4.8, we adjusted the weights for the different attributed. `Food` is set to 100 %, `accessories` to 75 % and `sauce` to 60 %.

The button `Set Similarity Configuration>>` gets the user to the next dialog.

### 4.4.3 The results

Illustration 4.9 shows the console output after the $i$ number of grapes were set in the similarity dialog. The console prints out the $i$ most probable grapes and their probabilities. This review was created with the final version of the system, and the grapes are therefore chosen on basis of the log odds ratio function instead of the probability. This is described in section 5.2.2. Figure 4.9 shows that the BN recommends Malbec, Cabernet Franc and Grenache to the lamb stew.



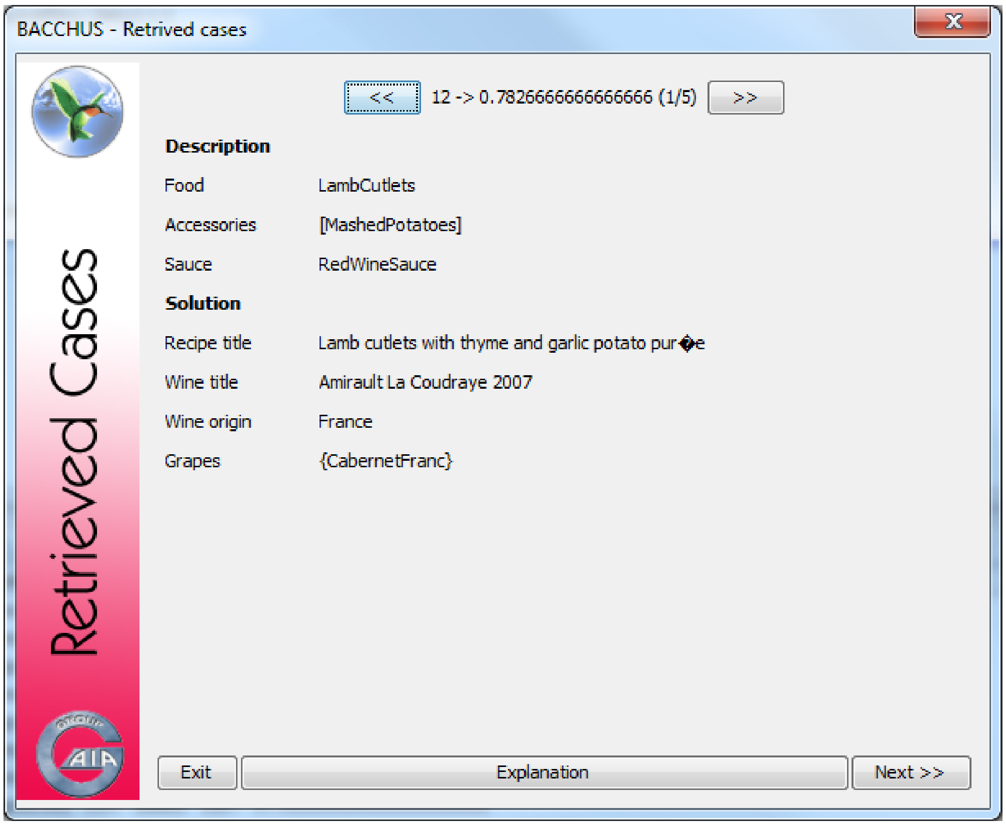
```
MealRecommender [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (24. juni 2010 16.50.41)
Values from the query:
FoodType: LambStew || SauceType: ProvenceSauce || AccType: MashedPotatoes

1. highest grape: Malbec with probability: 2.13 %  and log: 0.24
2. highest grape: CabernetFranc with probability: 6.99 %  and log: 0.16
3. highest grape: Grenache with probability: 8.68 %  and log: 0.15
```

Figure 4.9: The i best grapes.

After finding the most probable grapes, the retrieved cases are shown. This dialog is illustrated in figure 4.10. The best case match for the lamb stew was lamb cutlets. Between the two buttons in the top (i.e., << and >>) are some information about the case and the similarity. The first number is the case number (28), and then comes the case similarity after the arrow (0.78). The numbers in parentheses represents the position of this case among the other retrieved cases (i.e., this case is the first of the 5 that is retrieved).

The retrieved cases are divided into a description and a solution (i.e., the same division as described in section 4.1.1).

It is also possible to get an explanation of why the case was chosen by pushing the button `Explanation` in the lower middle.



| | |
|---|---|
| **Description** | |
| Food | LambCutlets |
| Accessories | [MashedPotatoes] |
| Sauce | RedWineSauce |
| **Solution** | |
| Recipe title | Lamb cutlets with thyme and garlic potato pur�e |
| Wine title | Amirault La Coudraye 2007 |
| Wine origin | France |
| Grapes | {CabernetFranc} |

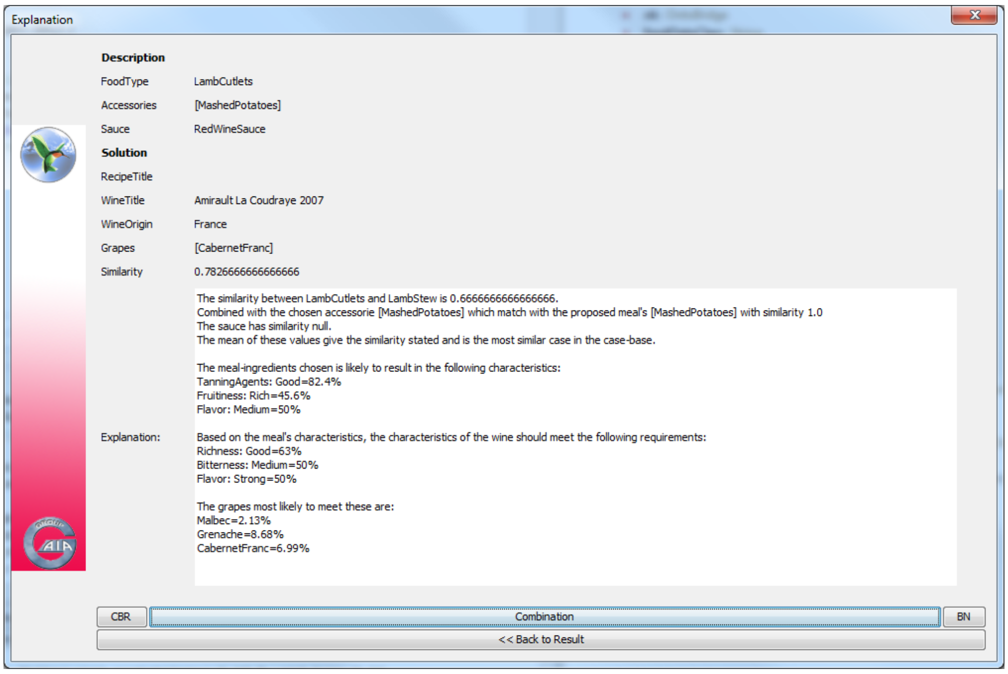
Figure 4.10: Result Dialog

### 4.4.4 The explanation

The explanation dialog shows up in a separate window from the result when the appropriate button is clicked in the result dialog.

There are three different explanations available from the three buttons shown in Figure 4.11: `CBR`, `Combination`, and `BN`. When either is clicked, the explanation available is shown in the explanation panel.

The `CBR` button gives an overview of the similarity values of the query case compared to the solution. `BN` gives three separate value types back; the first one is



**Description**

| | |
|---|---|
| FoodType | LambCutlets |
| Accessories | [MashedPotatoes] |
| Sauce | RedWineSauce |

**Solution**

| | |
|---|---|
| RecipeTitle | |
| WineTitle | Amirault La Coudraye 2007 |
| WineOrigin | France |
| Grapes | [CabernetFranc] |
| Similarity | 0.7826666666666666 |

Explanation:

```
The similarity between LambCutlets and LambStew is 0.6666666666666666.
Combined with the chosen accessorie [MashedPotatoes] which match with the proposed meal's [MashedPotatoes] with similarity 1.0
The sauce has similarity null.
The mean of these values give the similarity stated and is the most similar case in the case-base.

The meal-ingredients chosen is likely to result in the following characteristics:
TanningAgents: Good=82.4%
Fruitiness: Rich=45.6%
Flavor: Medium=50%

Based on the meal's characteristics, the characteristics of the wine should meet the following requirements:
Richness: Good=63%
Bitterness: Medium=50%
Flavor: Strong=50%

The grapes most likely to meet these are:
Malbec=2.13%
Grenache=8.68%
CabernetFranc=6.99%
```

Figure 4.11: The explanation dialog

the probabilities from the intermediary food tables, i.e., *Freshness*, *Bitterness*, and *Flavor*. The second group consists of the probabilities from the wine characteristics, while the last is the log-odds-ratio calculated for each of the `k` top grapes.

The middle button, `Combination`, simply combines these two into the same window.

The button `<<Back to Result` at the bottom simply closes the explanation window and returns the user the result dialog. If the user wish to peruse the other available solution suggestions, he may do so and use the same approach to get explanations for these in turn.

This is the last step implemented, so the `Next>>` button in the result dialog will take the user to a confirmation dialog that informs that the cycle is finished and asks if the user want to query again.

# Chapter 5

# Testing and results

Bacchus was tested in order to evaluate its utility value. This chapter elaborates how the testing was done and the results of the testing. The first section will describe how the tests was executed. Section 5.2 elaborates the testing of the Bayesian network. The network was first tested once, then revised and tested a second time. After this, we tested the whole system (Bacchus). This is described in section 5.3.

## 5.1 Testing environment

The system was tested in two different ways. First, we tested if the BN would retrieve the right grapes for a given meal. This was done first since the system depends on good results from the BN in order to produce good results itself. After the BN was tested, Bacchus were tested on the same meals. Since we are not qualified to decide if the results are applicable or not, we had to rely on experts to evaluate the answers.

We used a brochure named *Nyttig om mat og vin* from Vinmonopolet [55] and their web page to help us create the system tests. The brochure has a table with an overview of wine suggestions to over 100 meals. For each course, it describes the matching white wine, sparkling wine, red wine, fortified wine and their corresponding characteristics. The web page were used as supplementary information when we could not find a adequate entry in the brochure. In addition, we contacted Vinmonopolet at Solsiden in Trondheim and got one of their employees, Jorunn Hoøen to evaluate the results of the tests.

The executed tests are listed in table 5.1. The table was extended and adapted from Vinmonopolet [55] together with Jorunn Hoøen from Vinmonopolet. The course title and the input to the system does not necessary match. The course title is the same as the course title in the brochure, while the input is more specific. For example was course 4 extended with baked potatoes and Bearnaise sauce. The input was extended together with Hoøen in order to differentiate more between the different courses when testing on the BN. For example would course 1 and 2 be equal if the sauce were not set to be different.

## 5.2 Testing the BN

In order for the system to give correct answers, it is important that the BN returns the best grapes for the given meal. We started to test the BN to see if the network were able to correctly return the most adequate grapes in a overall manner. The tests wanted to prove if the network could find the right grapes for a number of classical courses. The system will typically use the three best grapes for further reasoning, and we want to find out if the BN would return Vinmonopolets suggestions as top three.

### 5.2.1 First test

The BN was first tested on the courses listed in table 5.1.

The result from the tests showed little distribution between the probabilities. The tests showed that the three grapes Cabernet Franc, Malbec and Tempranillo were top three in the system for five of the six first courses. As we can see from table 5.2, Cabernet Franc only matched Vinmonopolets suggestions in course 1, 2, 3 and 4 while Malbec and Tempranillo did not match the suggestions in any of the cases. Although, the BN finds one of the suggested grapes for five of the first six courses.

For course 7 - 13 (see table 5.3), the results are not as single-tracked. For these, the BN did suggest a total of 7 different grapes (i.e., Cabernet Franc, Malbec, Montepulciano, Pinot Noir, Sangiovese, Tempranillo and Tinta Miuda). Even though the distribution was better, the matching results were worse. For four of the courses (7, 10, 11 and 13), the BN could not find any of the suggested grapes. For course 9, it found one of the matching grapes, and for course 8 and 12, it found 2 matching grapes.

Tests showed that the BN managed to return at least one of the suggested grapes for 8 of the 13 courses. Although, the main problem with these results were that

| # | **Course title** | **Vinmonopolets grape suggestions** | **Input to Bacchus (Input to BN in parenthesis)** |
|---|---|---|---|
| 1 | Chicken, oven-baked with root vegetables | Pinot Noir, Barbera, Cabernet Franc | F: ChickenSteak(LightMeat), A: Vegetables S: NoSauce |
| 2 | Coq au wine | Pinot Noir, Syrah, Cabernet Franc | F: Cock(LightMeat), A: Vegetables, S: GravySauce |
| 3 | Beef/entrecôte/ steak from cattle | Chianti, Bordeaux | F: BeefTenderloin(Beef), A: BakedPotatoes(Potatoes) S: BearnaiseSauce |
| 4 | Pork, fillet | Cabernet Franc, Pinot Noir | F: PorkSirloin(Pork), A: Vegetables S: NoSauce |
| 5 | Pork rib | Sangiovese, Pinot Noir | F: PorkRib(Pork), A: BoiledPotatoes(Potatoes) S: GravySauce |
| 6 | Lamb, roast | Chianti, Tempranillo | F: LambRoast(Lamb_Sheep), A:MashedPotatoes (Potatoes), S: RedWineSauce |
| 7 | Norwegian lamb stew (Fårikål) | Valpolicella, Bardolino | F: Mutton(Lamb_Sheep), A: Cabbage(Vegetables), S: NoSauce |
| 8 | Bacalao | Barbera, Valpolicella, Portugeese wines | F: Clipfish(Fish), A: BoiledPotatoes(Potatoes), S: TomatoSauce |
| 9 | Halibut fried/grilled | Valpolicella, Pinot Noir | F: Halibut(Fish), A: Vegetables, S: CreamSauce |
| 10 | Gorgonzola | Valpolicella, Shiraz | F: Gorgonzola(Cheese), S: NoSauce |
| 11 | Hare, roast | Syrah, Zinfandel | F: Hare(SmallGame_Bird) A: Vegetables, S: GameSauce |
| 12 | Elk, roast | Syrah, Sangiovese | F: Elk(BigGame), A:Vegetables, S: CreamSauce |
| 13 | Greek salad | Cabernet Franc, Barbera | F: Salad(Vegetarian), S: NoSauce |

Table 5.1: Test table for system. The input to BN are put in parenthesis were different than input to the system. Some of the grape suggestions mentioned are special terms (e.g. Valpolicella). These terms are explained in Appendix B.

| | **Courses** | | | | | |
|---|---|---|---|---|---|---|
| **Grapes** | **1** | **2** | **3** | **4** | **5** | **6** |
| Barbera | **9.0 %** | 11.2 % | 11.3 % | 11.8 % | 12.1 % | 12.9 % |
| Cabernet Franc | ***9.1 %*** | ***11.8 %*** | **12.8 %** | ***12.3 %*** | *13.0 %* | *15.0 %* |
| Cabernet Sauvignon | 8.2 % | 10.7 % | **12.6 %** | 11.1 % | 11.7 % | 13.6 % |
| Carignan | 8.3 % | 10.7 % | 12.1 % | 10.6 % | 11.4 % | 13.6 % |
| Corvina | 6.7 % | 8.9 % | 11.2 % | 8.4 % | 9.1 % | 12.1 % |
| Grenache | 8.4 % | 11.2 % | 12.8 % | 11.3 % | 11.9 % | 14.4 % |
| Malbec | *9.1 %* | *12.0 %* | *14.4 %* | *12.3 %* | *13.1 %* | *16.0 %* |
| Merlot | 8.6 % | 11.1 % | **13.0 %** | 11.3 % | 12.0 % | 14.2 % |
| Montepulciano | 8.8 % | 10.8 % | 12.3 % | 11.3 % | 11.8 % | 13.5 % |
| Pinot Noir | **9.0 %** | **10.9 %** | 12.3 % | **11.3 %** | **11.8 %** | 13.4 % |
| Rondinella | 6.6 % | 8.7 % | 11.3 % | 8.1 % | 8.8 % | 12.0 % |
| Sangiovese | 8.8 % | 11.5 % | ***13.6 %*** | 11.7 % | **12.5 %** | ***15.0 %*** |
| Syrah | 8.0 % | **10.6 %** | 12.8 % | 10.6 % | 11.4 % | 14.2 % |
| Tempranillo | *9.4 %* | *11.9 %* | *13.4 %* | *12.6 %* | *13.3 %* | **14.9 %** |
| Tinta Miuda | 8.0 % | 9.4 % | 9.5 % | 10.6 % | 10.9 % | 10.3 % |
| Zinfandel | 7.7 % | 9.9 % | 12.2 % | 9.7 % | 10.4 % | 13.2 % |
| **Matches** | 1 | 1 | 1 | 1 | 0 | 1 |

Table 5.2: BN Results from first test with case 1 - 6. Vinmonopolets suggestions are marked in bold, while our systems top three are marked in italic. The last row represents the number of matching grapes suggestions (i.e., the number of top three grapes from our system that matches the grapes suggested by Vinmonopolet).

| | **Courses** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Grapes** | **7** | **8** | **9** | **10** | **11** | **12** | **13** |
| Barbera | 12.3 % | **7.9 %** | 7.7 % | 10.4 % | 11.0 % | 11.9 % | **5.1 %** |
| Cabernet Franc | 15.0 % | 8.4 % | 8.0 % | 13.6 % | *14.6 %* | *15.1 %* | **5.9 %** |
| Cabernet Sauvignon | 13.6 % | 7.8 % | 7.5 % | 12.7 % | 13.5 % | 14.0 % | 5.6 % |
| Carignan | 13.3 % | 7.6 % | 7.3 % | 11.6 % | 13.1 % | 13.6 % | 5.2 % |
| Corvina | **11.9 %** | **6.4 %** | **6.2 %** | **10.6 %** | 12.9 % | 13.2 % | 4.5 % |
| Grenache | 14.0 % | 7.6 % | 7.2 % | 12.2 % | 13.8 % | 14.4 % | 5.1 % |
| Malbec | *16.0 %* | 8.7 % | 8.2 % | *14.6 %* | *16.2 %* | *16.4 %* | 6.2 % |
| Merlot | 14.4 % | 8.5 % | 8.2 % | 13.3 % | 14.4 % | 14.5 % | 6.3 % |
| Montepulciano | 13.9 % | 9.0 % | *8.8 %* | 13.2 % | 13.6 % | 13.5 % | *6.9 %* |
| Pinot Noir | 13.8 % | *9.2 %* | ***9.1 %*** | 13.1 % | 13.4 % | 13.2 % | *7.2 %* |
| Rondinella | **11.9 %** | **6.5 %** | **6.3 %** | **10.8 %** | 13.2 % | 13.3 % | 4.8 % |
| Sangiovese | *15.2 %* | 8.7 % | 8.3 % | *14.0 %* | *15.3 %* | ***15.4 %*** | 6.5 % |
| Syrah | 14.1 % | 7.6 % | 7.3 % | **12.7 %** | **14.4 %** | **14.7 %** | 5.4 % |
| Tempranillo | *15.1 %* | ***9.1 %*** | 8.7 % | *13.9 %* | 14.3 % | 14.7 % | 6.8 % |
| Tinta Miuda | 10.9 % | ***9.0 %*** | *8.8 %* | 11.2 % | 9.7 % | 9.8 % | *8.1 %* |
| Zinfandel | 13.2 % | 7.5 % | 7.2 % | 12.1 % | **13.9 %** | 14.0 % | 5.4 % |
| **Matches** | 0 | 2 | 1 | 0 | 0 | 1 | 0 |

Table 5.3: BN results from first test with case 7 - 13. Vinmonopolets suggestions are marked in bold, while our systems top three are marked in italic. The last row represents the number of matching grapes suggestions. The last row represents the number of matching grapes suggestions (i.e., the number of top three grapes from our system that matches the grapes suggested by Vinmonopolet).

the probabilities were clustered. In order to get more significant results, we were advised by our co-supervisor Helge Langseth to revise our network.

Table 5.4 shows a summary of the results. These results are grouped by food type, in order to see how the BN perform on the different categories. The second column lists the courses with this food type. The third and fourth column lists how many of the courses that had one or more of Vinmonopolets wine suggestions as top 3 and 5 respectively. The BN has 62 % accuracy is we use the top 3 grapes, and 77 % accuracy is we use the top 5 grapes.

| **Food Type** | **Courses** | **Top 3** | **Top 5** |
|---|---|---|---|
| LightMeat | 1 and 2 | 2/2 | 2/2 |
| Beef | 3 | 1/1 | 1/1 |
| Pork | 4 and 5 | 1/2 | 2/2 |
| Lamb & Sheep | 6 and 7 | 1/2 | 1/2 |
| Fish | 8 and 9 | 2/2 | 2/2 |
| Cheese | 10 | 0/1 | 0/1 |
| SmallGame & Bird | 11 | 0/1 | 1/1 |
| BigGame | 12 | 1/1 | 1/1 |
| Vegetarian | 13 | 0/1 | 0/1 |
| **Sum** | 13 courses | 8/13 | 10/13 |

Table 5.4: Accuracy for BN model. The table shows how many courses of the different food type that got the suggested grapes as respectively top 3 and top 5.

### 5.2.2 Revision of BN

When we first created the BN, we normalized the probabilities within each grape. The sum of probabilities for each grape added up to 100%. This resulted in a clustered distribution of probabilities, since many wines have the same characteristics. As we can see from figure 5.1, all grapes have probabilities between 9% and 13%. This lack of span in the probabilities lead to trouble with choosing the right grapes to be used in further reasoning. In order to get greater distinctions between the results, we calculated new probabilities for each grape compared with the others. Instead of having each grape sum to 100%, we had each characteristic sum to 100%. For example did we find 15 bottles classified with *richness*: `medium`, *tanning agents*: `good` and *fruitiness*: `medium`. Of these were 6 bottles of Cabernet Sauvignon, 2 bottles of Carignan, 1 bottle of Grenache, 1 bottle of Merlot, 1 bottle of Montepulciano, 1 bottle of Sangiovese and 3 bottles of Tempranillo. Cabernet Sauvignon then got 40%, Carignan got 13%, Grenache got 7%, Merlot got 7%, Sangiovese got 7% and Tempranillo got 20%. The new probabilities are illustrated with the same evidence as in the first network (figure 5.1) in figure 5.2.

The new function used:

$$P(grape|characteristics) = \frac{\text{\# of bottles with characteristics for this grape}}{\text{total \# of bottles with characteristics}} \tag{5.1}$$

where *# of bottles with characteristics for this grape* is the number of bottles of the given *grape* and with the specific characteristics and *total # of bottles with this characteristic* is the total number of bottles having this characteristic (regardless of *grape*).

For Merlot, this would be:

$$P(Merlot|richness = medium, tanningAgents = good, fruitiness = medium) = \frac{\text{1 bottle with this characteristic}}{\text{15 bottles total for characteristics}} = 7\% \tag{5.2}$$

In order to discriminate even more between the grapes, we used the log odds ratio function. Log odds ratio is the logarithm of the odds ratio [9], which is the ratio between two probabilities. By finding the odds ratio, we can see if an event is more or less likely to occur in one group than another. In our case, we want to know if a grape is becoming more or less probable given the evidence (i.e., the food, sauce and accessories). The logarithm of the odds ratio is used to get an approximately normal distribution. Any values below 0 means that the grape is less likely to be suitable given the evidence. Values over 0 means that the grape is more likely to be adequate for the meal given the evidence. We are therefore interested in the log odds values over 0, and especially the three highest values (as we want to chose the three best grapes in our tests).

Log odds ratio equation:

$$L(O) = \log\left(\frac{P(grape|evidence)}{P(grape)}\right) \tag{5.3}$$

Note that the evidence in equation 5.3 is not the same as the characteristics in 5.1. While the numbers we inserted into the probability tables are directly dependent on the characteristics, the evidence used to calculate the log odds ratio are food, sauce and accessories type. This evidence gives us probabilities on the different characteristics. As we can see in figure 5.2, a meal containing Beef, Potatoes and Cream sauce has not one distinct result for each of the characteristics. The resulting probability for Barbera is therefore the sum of several characteristics.

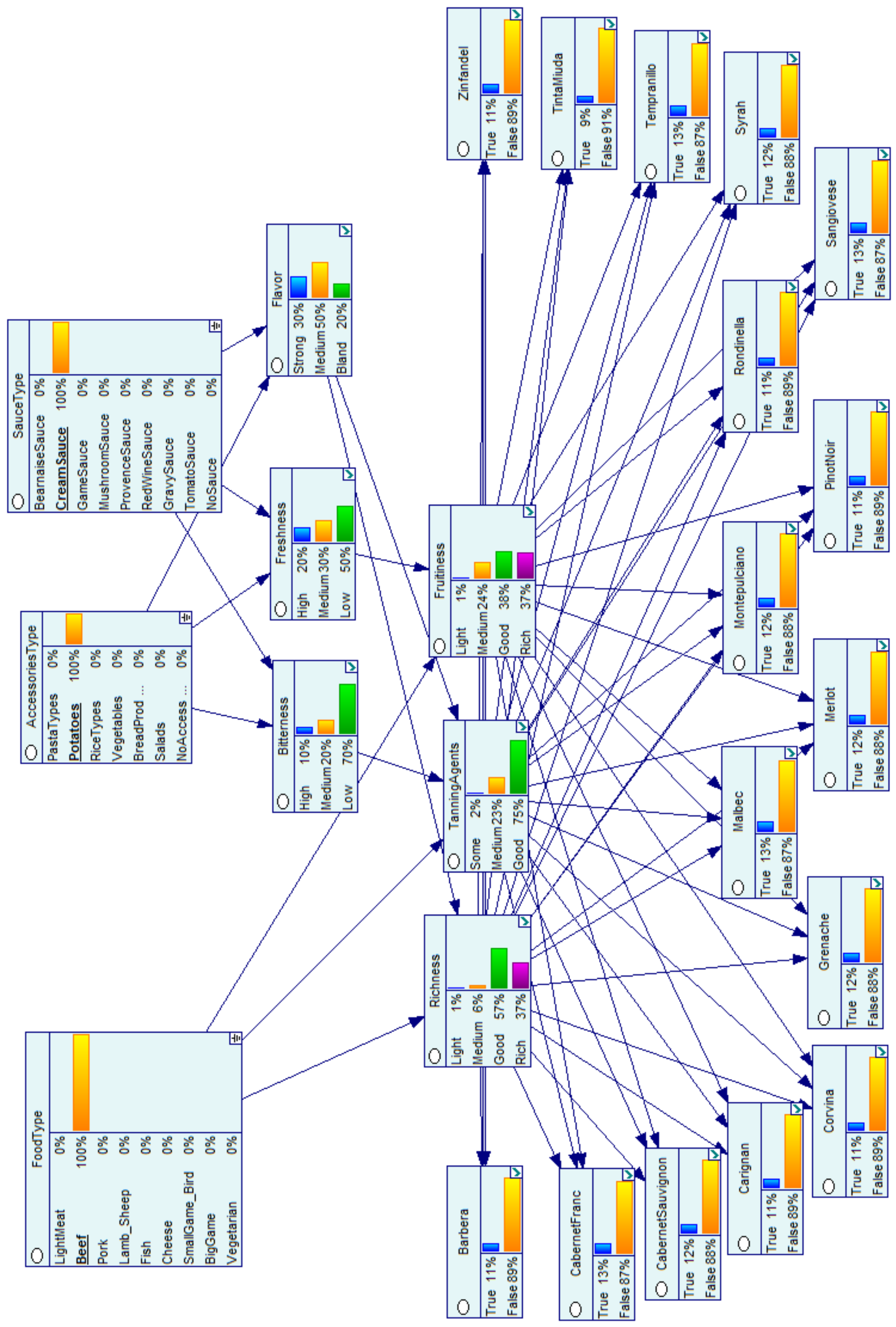


Figure 5.1: Grape values before the revision. The reader should notice the clustered probabilities.

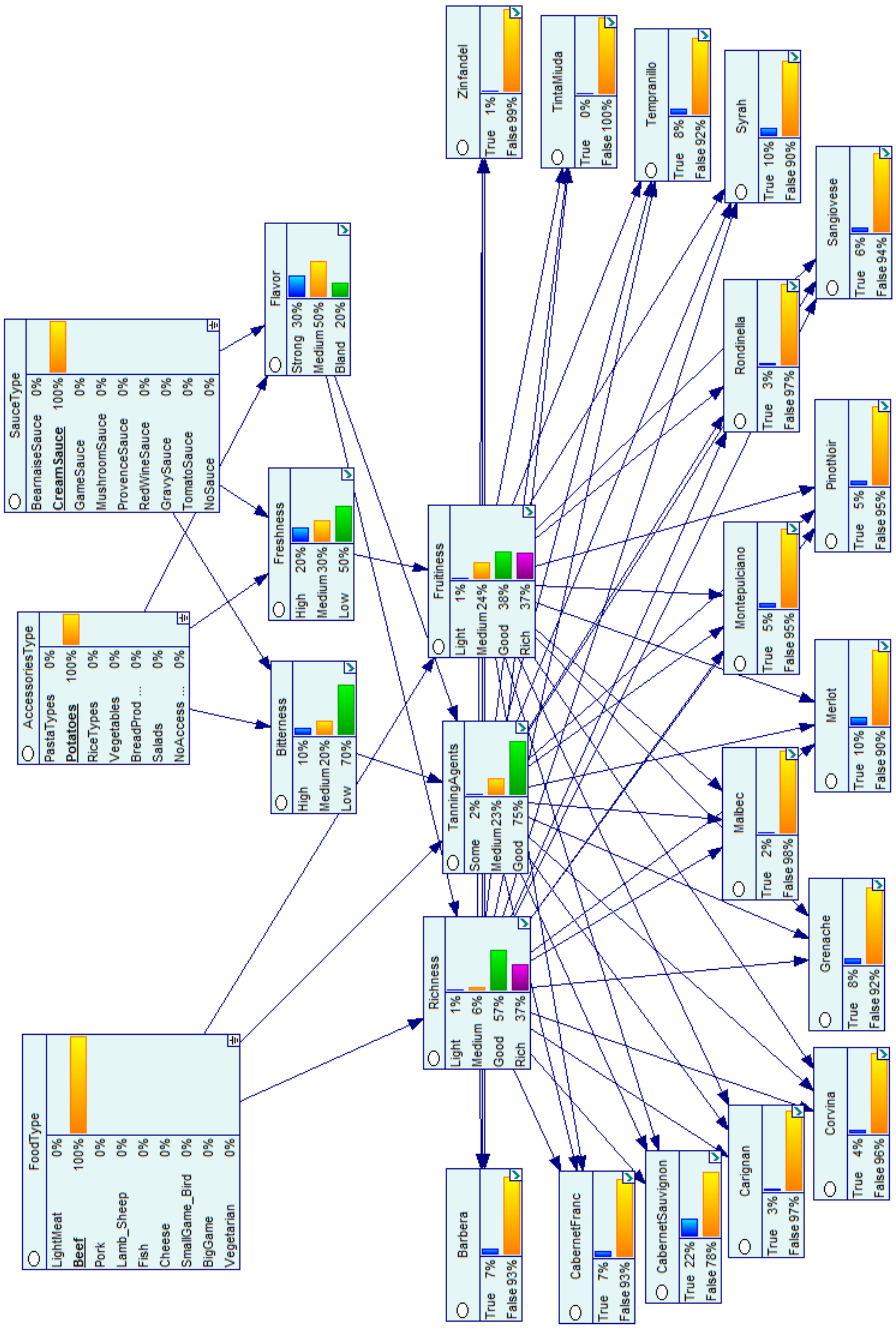


Figure 5.2: Grape values after the revision.

### 5.2.3 Second test

In the second test, we evaluated the results based on the log odds ratio values instead of the probabilities. The tables used in the section test of the BN is therefore formatted differently. Table 5.5 is a summary of the results found when testing the BN the second time.

| **Food Type** | **Courses** | **Top 3** | **Top 5** |
|---|---|---|---|
| LightMeat | 1 and 2 | 1/2 | 2/2 |
| Beef | 3 | 1/1 | 1/1 |
| Pork | 4 and 5 | 0/2 | 0/2 |
| Lamb & Sheep | 6 and 7 | 0/2 | 0/2 |
| Fish | 8 and 9 | 2/2 | 2/2 |
| Cheese | 10 | 0/1 | 0/1 |
| SmallGame & Bird | 11 | 1/1 | 1/1 |
| BigGame | 12 | 0/1 | 1/1 |
| Vegetarian | 13 | 0/1 | 0/1 |
| **Sum** | 13 courses | 5/13 | 7/13 |

Table 5.5: Accuracy for BN model. The table shows how many courses of the different food type that got the suggested grapes as respectively top 3 ang top 5.

As the table shows, 5 of the 13 meals had one or more of Vinmonopolets suggestions in top three, while 7 of the 13 courses had one of the suggestions in top five. These results are much worse than the results from testing the first BN (see tables 5.4), giving a top 3 accuracy of 28 % and a top 5 accuracy of 54 %. The table shows that the BN has problems finding the right grapes for Pork, Lamb, Cheese and Vegetarian. These courses had none of the suggested grapes as top five. One of the two LightMeat courses (course 2 - Coq) had one of the suggested grapes as number 4, and the BigGame course (course 12 - Elk) had two of the suggested grapes as number 4 and 5. The results from the tests shows that the BN has problems finding the right grapes for Pork, Lamb&Sheep, Cheese and Vegetarian dishes.

We have included one of the test tables showing the results from testing BN on course 6 - Bacalao (see table 5.6). The rest of the tables can be found in Appendix C.

Table 5.6 lists the grapes in their ranked order. Unlike the first BN test, we now ranked the grapes based on their log-odds ratio. The table shows that the BN managed to find 2 grapes that matched the suggestions from Vinmonopolet. The table shows that we have solved the problem with clustered grapes.

| **Course 8 - Bacalao** | | | |
|---|---|---|---|
| **Rank** | **Grapes** | **With Evidence** | **Log-Odds Ratio** |
| 1 | Tinta Miuda | 0.6 % | 0.600 |
| 2 | Montepulciano | 3.6 % | 0.521 |
| 3 | Rondinella | 4.9 % | 0.282 |
| 4 | Corvina | 6.1 % | 0.225 |
| 5 | Pinot Noir | 6.9 % | 0.121 |
| 6 | Cabernet Sauvignon | 21.3 % | 0.021 |
| 7 | Merlot | 9.7 % | -0.034 |
| 8 | Carignan | 3.2 % | -0.033 |
| 9 | Sangiovese | 5.8 % | -0.035 |
| 10 | Tempranillo | 7.3 % | -0.056 |
| 11 | Barbera | 6.7 % | -0.064 |
| 12 | Syrah | 8.3 % | -0.169 |
| 13 | Grenache | 6.3 % | -0.175 |
| 14 | Zinfandel | 0.4 % | -0.211 |
| 15 | Cabernet Franc | 4.4 % | -0316 |
| 16 | Malbec | 1.2 % | -0.378 |
| Matching suggestions | | 2 | |

Table 5.6: BN Results from second test. Vinmonopolet recommends Barbera, Valpolicella (mix of Corvina, Rondinella and Molinara) and Portuguese wines for this meal.

In order to get more feedback on the system, we got Jorunn Hoøen at Vinmonopolet to evaluate the results. Hoøen claimed that the results could not be precluded from being acceptable, because almost all grapes could be modified to get different characteristics in a wine. Both the combinations of grapes and the growth conditions are among the elements that affects the outcome. This showed some of the poor aspects of the model. Because the model was made very general, it does not include such information. In addition, it lacks the ability to differentiate between important aspects as how the meal was prepared (i.e. if the ingredients were cooked or fried), and potential herbals etc. used in the meal. A meal with raw vegetables are completely different from a meal with cooked vegetables. Because of this generality in the model, we could not conclude that the results from the BN were good or bad. Hoøen also told us that Vinmonopolets recommendations were biased towards known grapes with a larger assortment available. Our model represents both known grapes and some not so known grapes. This could to some extent explain our results.

## 5.3 Testing Bacchus

In order to test if the system produces acceptable results, we had to test it as a whole. The system were tested on the 12 of the courses we tested the BN on (see table 5.1). Unfortunately, at the day of the testing, we noticed that the results for course 1 was incorrect and these results are therefore only evaluated by our self and disregarded in the results. All tests were executed with the same similarity configurations:

- Similarity functions
  - Food: OntDeepBasic
  - Accessories: OntDeepBasic
  - Sauce: myCBRTableSim
- Similarity weights
  - Food: 100%
  - Accessories: 75%
  - Sauce: 60 %
- Selecting 3 best grapes
- Returning 5 best cases

We were lucky enough to get Jorunn Hoøen to evaluate the whole system. She studied the results for each of the test courses and commented each of the wine suggested by the system.

The test of the BN showed that it had problems picking the most adequate grapes for dishes with lamb and sheep. We were therefore interested in how the whole system would do on these courses. Table 5.7 shows the results of testing the system on course 7, ”Fårikål” (Norwegian lamb stew). As we can see, the system managed to find an exact match in the case base. Domaine Robert Vic Cuvée Sélection 2008 is made of a mix of Merlot, Syrah and Grenache, which is none of the grapes recommended by Vinmonopolet (i.e., Valpolicella and Bardolino). This supports the fact that a wine could be adequate for a meal even though it does not have the ”right” grapes. The other wines suggested are used in very different meals, ranging from cod, pork and beef to cheese. This may seem strange for most people. Could a wine that were used in a course with steamed cod be working with ”Fårikål”? Hoøen actually claims that four of these five wines can perfectly well be used to a ”Fårikål” meal. The only wine she would not recommend is Argento Malbec 2008/2009, as she claims that this wine is a little bit too heavy for this course.

| **Course 7 - ”Fårikål”** | | | |
|---|---|---|---|
| **#** | **Similarity** | **Wine Suggestion** | **Used in course** |
| 1 | 1 | Domaine Robert Vic Cuvée Sélection 2008 | Norwegian lamb stew |
| 2 | 0.63 | Oléa Signature 2008 | Steamed cod on a bed of lentils with vegetables |
| 3 | 0.63 | La Trincherina Barbera d’Asti 2007 | Exotic pork fillet |
| 4 | 0.57 | Elsa Malbec 2005 | Beef Pebre |
| 5 | 0.52 | Argento Malbec 2008/2009 | Cheese table |

Table 5.7: Test results for course 7. The first four wines were claimed to be very good matches, while the last one was too heavy.

Hoøen were positive to our results, and said that most of the wines was really good suggestions. The only time the system did not manage to suggest adequate wines, was for course 2 - Coq au wine. The results for this course are listed in table 5.8. According to Hoøen, this course needs juicy wines, and the resulting wines were too heavy.

All in all, Hoøen judged 52 of the 60 suggested wines as adequate, which gives us an accuracy of 87 %. Except from course 2, which was complete miss, the system

| **Course 2 - Coq au wine** | | | |
|---|---|---|---|
| **#** | **Similarity** | **Wine Suggestion** | **Used in course** |
| 1 | 0.73 | Isole e Olena Chianti Classico 2004 | Elk stew |
| 2 | 0.73 | Gran Coronas Reserva 2005 | Stewed beef |
| 3 | 0.59 | Wolf Blass Yellow Label Cabernet Sauvignon 2003/2004 | Elk steak |
| 4 | 0.59 | Villa Antinori 2005 | Baked calf fillet |
| 5 | 0.57 | Carmen Cabernet Sauvignon 2004 | Roedeer fillet |

Table 5.8: Test results for course 2. None of these wines are adequate for this course.

had 3 suggestions that were not acceptable divided among course 5, 7 and 10. Hoøen declared that the system had an overall good performance, and she said that the results for course 3 (Beef), 9 (Halibut) and 11 (Hare roast) was very good.

Complete tests can be found in appendix C.

# Chapter 6

# Discussion

In this chapter, we will analyze our results and discuss if our initial thesis objectives was fulfilled. The first goal of this project was to further specialize and implement the architecture we proposed in the previous specialization project [18]. Then should this implementation and its utility value be evaluated. This architecture has been implemented and tested in chapter 4 and 5, and these results will now be discussed. In the first two sections, we will evaluate the strength of the Bayesian Network and the case base. In section 6.3, we will discuss the strength of the system (Bacchus). The choice of technology will be evaluated in 6.4, before we explain how and why our design can be utilized in the TLCPC project in section 6.5. The last section will summarize further work that should be done.

## 6.1 The strength of the BN

The network was mainly built on knowledge from the Norwegian Vinmonopol. Vinmonopolet has quantified information about the coherence between food types and wine characteristics and grapes. This was utilized in the creation of the BN, but we could not find equivalent information about the coherence between the accessories and sauce and the characteristics. Hence, we had to use guidelines from Vinmonopolet [54] to quantify these links. This part of the BN was therefore quantified based on our own assessments, and could therefore not be regarded as fully reliable.

Because of the large amount of work necessary to fill out the probability tables, we chose 16 grapes to represent in our system. The grapes were chosen in parallel with finding cases in the case base, and they consists of the grapes used in the courses in the case base. The grapes represented are therefore a selection of the most known grapes and some lesser known ones. We encountered a challenge when we wanted to represent a grape that has different names in different countries (i.e., Tempranillo is called Tinta Roriz in Portugal and Syrah is called Shiraz in Australia). We decided to use the most used names (i.e., Tempranillo and Syrah), and thus classify Tinta Roriz wines as Tempranillo wines and wines with Shiraz as Syrah wines.

We decided to only represent single grapes, not grape mixes. In addition to be classified by grape mixes, wines can also be classified by their region, since the climate and geography affects the taste. Bordeaux is a term for all wines produced in the Bordeaux region in France. In our system, a Cabernet Sauvignon from Bordeaux will be equal to a Cabernet Sauvignon from Italy or California. We also have the Rondinella and Corvina grapes that are blended to get Valpolicella or Bardolino wines. These grapes should probably be represented as a mix instead of individually. Another reason for this is the way we filled out the probability tables. These numbers was found by quantifying the results for a specific grape on their web page (`http://www.vinmonopolet.no/`). A search for Rondinella will currently give us 165 hits, while a search for Valpolicella gives us 141 hits. This means that 85 % of the wines with the Rondinella grape are Valpolicella mixes. 72 % of the Corvina wines are also Valpolicella mixes. Which brings us to how the results are represented by Vinmonopolet. A search for a grape will return all wines having this grape in it, independent of the amount. This means that the probability for a grape will be affected with what grape combinations it is used in. For Rondinella, this is specially evident, as 85 % of the wines it occur in, is Valpolicella wines. In addition, Rondinella is also outnumbered by Corvina, which usually represents 40 - 70 % of the content, while Rondinella only represents 20 - 40 %. As the BN is today, the Rondinella grape is therefore dedicated characteristics that may come from the Corvina grape. We could expect that this would lead to clustered results for these two grapes in the BN. The tests showed that these grapes were following each other in the result list in only 4 of the 13 tests done on the second version of the BN. One can also question if the grapes should be represented only as itself, only as mixes or as both. By representing them only as them self, they get influenced by other grapes that is present in the bottles. By representing them only as mixes, they would loose their individual characteristics. We think that the best way is to represent the grapes as both mixes and as individuals. This would probably require a full revision of the BN, as we would need to represent the grapes characteristics individually and this is problematic to do by using Vinmonopolets assortment lists.

We were lucky enough to get Vinmonopolet to evaluate the BN. As suspected, Jorunn Hoøen said that the model was too general to be able to return the best grapes for every course. In order to get better results, the model should therefore be more specialized. Due to time constraints, we did not focus on this aspect in our work. Hoøen recommended that we also represented countries in connection with the grapes and more details about the food (e.g., how it is prepared, special herbals etc.). This is left as further work. Hoøen was also critical to us representing Tinta Roriz as Tempranillo and Syrah as Shiraz. She claimed that the Syrah grape was different in Australia. As we have created the BN with use of the characteristics of the Syrah grape, it would not be correct to classify the Shiraz grape as a Syrah grape. If this should be done, the Syrah node in the BN should therefore be updated with values representing both Syrah and Shiraz. The same applies to the Tempranillo grape, which should be updated with values representing both Tempranillo and Tinta Roriz.

The structure of the BN was created to facilitate extensions on the model. Our model has 3 input nodes, and 16 output nodes. Between these two, there are also two layers of intermediate nodes. These two intermediate layers can be extended without having any impact on the system. When we first extended our system by adding the three nodes Bitterness, Freshness and Flavor, no additional code was needed for the system to run.

## 6.2 The strength of the case base

The case base is based on courses found at Apéritifs web page. We have included only the most essential ingredients, as many of the courses have a number of different accessories. For example has the course Bacalao from Lofoten the accessories potatoes, onion, tomatoes, peppers and bread in Apéritifs meal database. In our system, we chose to represent Bacalao from Lofoten as a dish with boiled potatoes and bread as accessories. We chose to only use the most essential ingredients because we believed that these ingredients will have the most impact on the matching. The tomato sauce and the potatoes will be more significant for the result than for example the onion. If we extended the accessories lists, we would probably have to create new similarity measures that distinguish between essential accessories and others as onion, butter etc. Most meals in our case base have 2 associated accessories, but some also have 4 or 5. According to this, it could have been interesting to test if the amount of accessories have an influence on the results.

A case in our system presents one wine suggestion. As many of the meals at Apéritif has several wine suggestions, we could have added support for multi-valued wine lists. This functionality was omitted in order to not over complicate the system, and is left as further work.

The case base has four attributes representing the meal (i.e., recipe title, food type, accessories type and sauce type), and 3 attributes representing the wine (i.e., wine title, wine origin and grape type). The attributes for the meal are kept to a minimum. We chose to add the course title, since it often gives the reader more information than the generalized contents. The title adds information about how the meal was prepared (i.e., that the pork is whirled in bacon, or that the lamb cutlets comes with thyme and garlic potato purée). Regarding the wine attributes, we chose to add wine region in addition to the two obvious (i.e., wine title and grape type). The region attribute was added because we wanted to allow for it to be used in an eventual further development of the system.

It could have been interesting to see if having wine characteristics in a case would bring an added value. This could have been connected to the characteristics in the BN, showing this wines characteristics in proportion to the suggested characteristic.

If we look at the system from a user perspective, it could have been useful to have links to the courses represented in the cases. As this is not relevant for our work, this was omitted.

## 6.3 The strength of the system

The system manages to incorporate BN and CBR in a manner that utilizes both methodologies. The BN adds general knowledge to the system, making it possible to find new solutions based on the characteristics of a meal.

By using the BN to find the best grapes according to the characteristics of the meal, we are able to chose a wine based on the different flavors present and not only the specific food. According to Marthinsen and Hansen [35], are the food (i.e., meat, fish, cheese etc.) often less important than the sauce. There are also a relation between the amount of fat in the course and the amount of alcohol in the wine. The more fat, the more alcohol can the wine have. This also suggests that we should add a relation between these in our system.

Our system although have the limitation of depending on an initial case base to reason. The BN are of no use if the case base are empty, since it is used as part of the reasoning process in CBR, and not as a standalone reasoning process. This is not a big problem in the food domain, as there are plenty of recipes available at Apéritif. We also found that many systems uses a BN created from the data in the case base [13, 20, 45, 51, 52]. There is reason to believe that our system profit on having an independent BN created from expert knowledge.

As we can see from the results from testing the system on course 9 (see 6.1), the system managed to return 3 cases with Valpolicella wines, which are the grape suggestion from Vinmonopolet. Of these courses, one contains coalfish, another turkey and the last pork. In addition, the turkey meal and the pork meal has the exact same wine suggestion, Zenato Valpolicella Classico Superiore 2007. We also got the system evaluated by a wine expert, Jorunn Hoøen at Vinmonopolet, which classified the results for this course as very good. These results support our hypothesis that the BN could bring an added value to the system.

| **Course 9 - Halibut, fried/grilled** | | | |
|---|---|---|---|
| **#** | **Similarity** | **Wine Suggestion** | **Used in course** |
| 1 | 0.48 | Duca del Frassino Valpolicella Superiore 2007 | Baked coalfish |
| 2 | 0.45 | Vidigal Reserva 2005 | Coq au wine |
| 3 | 0.45 | Zaccagnini Montepulciano d'Abruzzo 2005 | Pheasantbreast with olive-gravy |
| 4 | 0.45 | Zenato Valpolicella Classico Superiore 2007 | Turkey fillet on a bed of vegetables |
| 5 | 0.45 | Zenato Valpolicella Classico Superiore 2007 | Pork rib |

Table 6.1: Test results for course 9. The complete result table is found in Appendix C

We chose to use the top three grapes as the standard number of grapes to be used for further reasoning. Hoøen questioned this, as she meant that it would be beneficial to use more grapes. A user could therefore easily change the amount of grapes in the systems GUI. Our accuracy of 87 % bear witness that our system can in fact return adequate cases by using only the top three grapes.

As the system is now, a user can only pick one food type, one accessory and one sauce in the meal query (see 4.4.1). We wanted to test the system with basic functionality at first. Multi-values query objects can be implemented later and tested to see if this yields better results. If the system should have the possibility to choose several accessories, there might be necessary to separate essential accessories and less essential accessories. The most essential accessories should always have a greater influence on the similarity.

In the architecture we proposed in the previous specialization project, we included clinical guidelines to represent general knowledge. As there exists a lot of guidelines used to match meals and wines, this system could probably beneficially be extended with guidelines. We picture the guidelines as rules linking characteristics in a meal to characteristics in the wine (i.e., if the meal has a fatty content it could be matched with a wine with higher alcoholic strength).

Our system has proven that it can solve the problem with competence gaps (mentioned in section 1.1.3). By utilizing the BN, our system can manage to retrieve wine suggestions that match the input case, even though the cases differ. As our system chooses results based on characteristics, it is not essential to have a vast case base that covers all types of food.

### 6.3.1 Similarity Functions

In our system, we used some of the similarity functions incorporated in jColibri (i.e. OntDeepBasic, OntDeep, OntCosine, OntDetail) to deal with the ontologies in our system. In addition, we created our own similarity function with myCBR that was used on sauce. Empirical tests done during the implementation favored OntDeepBasic as the best jColibri similarity function to use on the ontology based ingredients. Although, even though the different similarity functions returns different similarities, the resulting cases are mostly the same.

We could also have created myCBR similarity functions for the ontologies. This requires more work than using the built-in functions, but could have offered more precise results. This is left as further work. An thorough study of different similarity measurements should be carried out, but was outside our scope.

## 6.4 Choice of technology

Some of the technologies we chose were good tools that managed to add value to our system, while others turned out to difficult to work with.

Protégé was used to create the ontologies, which were easy to create and export to jColibri. myCBR is a plug in to Protégé, which were straight forward to prototype the CBR part of our system with myCBR and Protégé. One particular benefit of myCBR is that we can create similarity functions that can be used in jColibri.

The first problem we ran into was Hibernate. We were not able to get Hibernate to work with HSQLDB. In order to avoid loosing to much time, we switched to Apache Derby. This helped us some, but we still encountered a lot of problems before we got Hibernate to work properly. An extensive amount of error messages was encountered and fixed. Hibernate was chosen because it simplifies the transactions to and from the database, and it relieved us for a lot of work once we got it to work. But all in all, we believe that we used too much time in order for it to work. These problems was partially caused by jColibri, as we felt that the tutorial [44] was insufficiently explaining how jColibri, the database and hibernate could be mapped together.

We also ran into other problems with jColibri. Because of its sophisticated design, it is relatively complex [5]. And, as many other frameworks, it was insufficiently documented. A lot of time went by on debugging. On the other side, jColibri is an advanced framework, and has a lot of functionality available that can be utilized.

GeNIe was used to create the Bayesian network. This interface considerably simplified the work of creating this model. The problems began when we wanted to use the similarities in our system with the help of SMILE. Networks created in GeNIe are saved as XML files. The structure of this files makes it difficult to update a special value for a given node (e.g., get probability for node Richness with state light and parents FoodType = LightMeat and Flavor = Medium). All values for a node are stored in a list, and it is complicated to find the exact right item in the list. We also experienced problems with the methods in SMILE and having to take long detours to get the information we wanted from the network.

## 6.5 TLCPC and our design

Our design is based on the wine and food domain, because the medical data from the TLCPC project was delayed. Our experiences from developing and testing this system can be utilized when creating the TLCPC system.

TLCPC aims to develop a decision support system that can be used for classification and treatment of pain. In this system, the project team plan to use the BN to represent generalized and insecure medical information. Our system has successfully shown that it can utilize the general and insecure information represented in our BN. There is thus reason to believe that our approach is transferable to the medical domain.

Our design are based on using ontologies to represent general concepts and their relationships. As we mentioned in section 2.4, the Unified Medical Language System (UMLS) gives us an ontology for the medical domain. This vocabulary includes categories as diagnosis, procedures, drugs, diseases etc of Medicine [42]. CARE-PARTNER [6–8] is one of the systems that use UMLS to represent relations between knowledge. This ontology can be utilized to represent medical information in a further development of our architecture in the medical domain.

In this thesis, we have shown a possible way to integrate CBR and BN that yielded good results when we tested the system on a wine expert. These results support that our integration can bring the TLCPC system an added value.

## 6.6 Further work

There are several issues that should be dealt with in a further development of the system. First of all, the BN should be revised with an wine expert. As mentioned in section 6.1, Jorunn Hoøen claimed that the model was to general. The whole BN should be revised and the representation of the grapes should be examined. The probabilities in the accessory and sauce nodes are currently not quality assured, and should be reviewed together with the rest of the BN.

SMILE was experienced as a bit difficult to work with. It could therefore be considered looking for another framework. The case base could also have been revised. First of all could the attributes be evaluated. Support for multi-valued attributes in the MealDescription (i.e., the input variables) might also strength the system. A case could also be extended to include characteristics for the suggested wine (e.g., the characteristic charts used by Vinmonopolet).

Our system could also benefit from having its own guidelines representing how food and wine should be combined. This could be done by adding a rule-based reasoning module to the existing system.

As mentioned, we used the jColibri similarity functions for the attributes defined by ontologies. It would probably be beneficial to create our own similarity functions. There should be done a thorough evaluation of the similarity functions to find out which of the functions yields the best results.

The ontologies in our system are relatively flat, and should be revised. The ontologies have a tight bond to the BN, and a revision of either the BN or the ontologies should therefore be suited to both the BN and the ontologies. Several of the classes in the ontologies should have defined more subclasses. For example is BigGame the class of the instances Deer, Elk, Reindeer and Roedeer. These could instead have been subclasses with their own instances.

We hope that someone will continue to work on our system and pass on the system to future students.

# Chapter 7

# Conclusion

In this thesis, we have presented a decision support system that combines Case-based reasoning and Bayesian networks. This system was first designed to be used in palliative care, but was developed with food and wine data because of delays in the preparation of the medical datasets. In our work, we have focused on developing an integration of CBR and BN that can utilize these methodologies' strengths.

Our system uses a Bayesian network to represent general relationships between food and grapes. In addition, we have a independent case base that represents recipes. By using the Bayesian network to reduce the search space for the case-based reasoning module, we were able to find new and interesting solutions in our system. Our design utilizes the strengths of the two reasoning methodologies and are thus able to overcome the single approaches limitations.

The most apparent problem with our system is the Bayesian network. This network was claimed to be too general by Jorunn Hoøen, a wine expert at Vinmonopolet, and should therefore be revised.

Our system has been tested and evaluated by an wine expert in the task of finding an adequate wine for a given meal, and we were able to get an accuracy of 87 % on the 12 cases we tested the system on. Our system is still under development, but we believe that a full implementation of our design could improve decision making for both the wine domain and the medical domain.

# References

[1] A. Aamodt and H. Langseth. Integrating bayesian networks into knowledge-intensive cbr. In *American Associationfor Artificial Intelligence, Case-based reasoning integrations*, pages 1–6, 1998.

[2] A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1):39–59, 1994.

[3] Apéritif. Apéritif, 2010. `http://www.aperitif.no/`.

[4] J. Arcos, R. De Mantaras, and X. Serra. Saxex: A case-based reasoning system for generating expressive musical performances. *Journal of New Music Research*, 27(3):194–210, 1998.

[5] J. J. Bello-Tomas, P. A. Gonzalez-Calero, and B. Diaz-Agudo. Jcolibri: An object-oriented framework for building cbr systems. *Advances in Case-Based Reasoning, Proceedings*, 3155:32–46, 2004.

[6] I. Bichindaritz. Solving safety implications in a case based decision-support system in medicine. *Proceedings of the 5th International Conference on Case-Based Reasoning*, 2003.

[7] I. Bichindaritz, E. Kansu, and K. Sullivan. Case-based reasoning in care-partner: Gathering evidence for evidence-based medical practice. In *Advances in Case-Based Reasoning*, volume Volume 1488/1998 of *Lecture Notes in Computer Science*, pages 334–345. Springer Berlin / Heidelberg, 1998.

[8] I. Bichindaritz, M. F. Siadak, J. Jocom, C. Moinpour, E. Kansu, G. Donaldson, N. Bush, M. Chapko, J. M. Bradshaw, and K. M. Sullivan. Care-partner: a computerized knowledge-support system for stem-cell post-transplant long-term follow-up on the world-wide-web. *Proc AMIA Symp*, pages 386–90, 1998.

[9] J. M. Bland and D. G. Altman. Statistics notes: The odds ratio. *BMJ*, 320 (7247):1468–, 2000.

[10] L. Branting. Integrating cases and models through approximate-model-based adaptation. 1998.

[11] A. Caraceni, N. Cherny, R. Fainsinger, S. Kaasa, P. Poulain, L. Radbruch, and F. De Conno. Pain measurement tools and methods in clinical research in palliative care: recommendations of an expert working group of the european association of palliative care. *J Pain Symptom Manage*, 23(3):239–55, 2002.

[12] G. A. Diamond and S. Kaul. Bayesian classification of clinical practice guidelines. *Arch Intern Med*, 169(15):1431–5, 2009.

[13] T. Dingsoyr. Retrieval of cases by using a bayesian network. 1998.

[14] K. Elvidge. Improving pain & symptom management for advanced cancer patients with a clinical decision support system. *Stud Health Technol Inform*, 136:169–74, 2008.

[15] A. Fornells, J. Recio-Garcia, B. Diaz-Agudo, E. Golobardes, and E. Fornells. Integration of a methodology for cluster-based retrieval in jcolibri. *Case-Based Reasoning Research and Development*, pages 418–433, 2009.

[16] A. X. Garg, N. K. J. Adhikari, H. McDonald, M. P. Rosas-Arellano, P. J. Devereaux, J. Beyene, J. Sam, and R. B. Haynes. Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: A systematic review. *JAMA*, 293(10):1223–1238, 2005.

[17] J. H. Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubezy, H. Eriksson, N. F. Noy, and S. W. Tu. The evolution of protege: an environment for knowledge-based systems development. *International Journal of Human-Computer Studies*, 58(1):89–123, 2003.

[18] A.M. Gravem. Cbr in clinical decision support for assessment and treatment of pain, 2009.

[19] G. W. Hanks, F. Conno, N. Cherny, M. Hanna, E. Kalso, H. J. McQuay, S. Mercadante, J. Meynadier, P. Poulain, C. Ripamonti, L. Radbruch, J. R. Casas, J. Sawe, R. G. Twycross, and V. Ventafridda. Morphine and alternative opioids in cancer pain: the eapc recommendations. *Br J Cancer*, 84(5):587–93, 2001.

[20] D. N. Hennessy, B. G. Buchanan, and J. M. Rosenberg. Bayesian case reconstruction. *Advances in Case-Based Reasoning*, 2416:148–158, 2002.

[21] D. L. Hunt, R. B. Haynes, S. E. Hanna, and K. Smith. Effects of computer-based clinical decision support systems on physician performance and patient outcomes: A systematic review. *JAMA*, 280(15):1339–1346, 1998.

[22] M. E. Johnston, K. B. Langton, R. B. Haynes, and A. Mathieu. Effects of computer-based clinical decision support systems on clinician performance and patient outcome: A critical appraisal of research. *Annals of Internal Medicine*, 120(2):135–142, 1994.

[23] M. S. Jordhøy, P. Fayers, J. H. Loge, M. Ahlner-Elmqvist, and S. Kaasa. Quality of life in palliative cancer care: Results from a cluster randomized trial. *Journal of Clinical Oncology*, 19(18):3884–3894, 2001.

[24] S. Kaasa and F. De Conno. Palliative care research. *European Journal of Cancer*, 37:S153–S159, 2001.

[25] H. Knublauch, R. W. Fergerson, N. F. Noy, and M. A. Musen. The protege owl plugin: An open development environment for semantic web applications. *Semantic Web - Iswc 2004, Proceedings*, 3298:229–243, 2004.

[26] P. Koton. Reasoning about evidence in causal explanations. volume 256, page 261, 1988.

[27] Decision Systems Laboratory. Decision systems laboratory wiki, 09.02.2010 2009. `http://genie.sis.pitt.edu/wiki/Main_Page`.

[28] M. B. Landrum and S.-L. T. Normand. Applying bayesian ideas to the development of medical guidelines. *Stat Med*, 18(2):117–37, 1999.

[29] P. Lucas. Knowledge acquisition for decision-theoretic expert systems. *AISB QUARTERLY*, pages 23–33, 1996.

[30] M. Maltoni, A. Caraceni, C. Brunelli, B. Broeckaert, N. Christakis, S. Eychmueller, P. Glare, M. Nabal, A. Vigano, P. Larkin, F. De Conno, G. Hanks, and S. Kaasa. Prognostic factors in advanced cancer patients: evidence-based clinical recommendations–a study by the steering committee of the european association for palliative care. *Journal of Clinical Oncology*, 23(25):6240–8, 2005.

[31] S. Mani and M. J. Pazzani. Guideline generation from data by induction of decision tables using a bayesian network framework. *Proc AMIA Symp*, pages 518–22, 1998.

[32] C. Marling, E. Rissland, and A. Aamodt. Integrations with case-based reasoning. *The Knowledge Engineering Review*, 20(03):241–245, 2006.

[33] C. Marling, J. Shubrook, and F. Schwartz. Toward case-based reasoning for diabetes management: A preliminary clinical study and decision support system prototype. *Computational intelligence*, 25(3):165–179, 2009.

[34] C. R. Marling. Integrating case-based and rule-based reasoning to meet multiple design constraints. *Computational intelligence*, 15(3):308, 1999.

[35] T. Marthinsen and H. I. Hansen. *Se lukt smak - håndbok i vinsmaking.* Vigmostad & Bjørke, 1. edn edition, 2008.

[36] S. Montani. Exploring new roles for case-based reasoning inÂ heterogeneous ai systems for medical decision support. *Applied intelligence*, 28(3):275–285, 2008.

[37] S. Montani and R. Bellazzi. Exploiting multi-modal reasoning for knowledge management and decision support: an evaluation study. Proceesings of the AMIA Symposium, page 585. American Medical Informatics Association, 2000.

[38] S. Montani and R. Bellazzi. Supporting decisions in medical applications: the knowledge management perspective. *International journal of medical informatics*, 68(1-3):79–90, 2002.

[39] S. Montani, R. Bellazzi, L. Portinale, and M. Stefanelli. A multi-modal reasoning methodology for managing iddm patients. *International journal of medical informatics*, 58:243–256, 2000.

[40] S. Montani, P. Magni, R. Bellazzi, C. Larizza, A. V. Roudsari, and E. R. Carson. Integrating model-based decision support in a multi-modal reasoning system for managing type 1 diabetic patients. *Artificial intelligence in medicine*, 29(1-2):131–151, 2003.

[41] NTNU. Translational research in lung cancer and palliative care - from genomics to symptom control (tlcpc), 2008.

[42] National Library of Medicine. Unified medical language system (umls), 2010. `http://www.nlm.nih.gov/research/umls/`.

[43] L. Radbruch and F. Nauck. [morphine and alternative opioids in cancer pain: the eapc recommendations]. *Schmerz*, 16(3):186–93, 2002.

[44] J. A. Recio-Garcia, B. DÃaz-Agudo, and P. GonzÃ¡lez-Calero. jcolibri2 tutorial. Technical report, Department of Software Engineering and Artificial Intelligence, 2008.

[45] A. F. Rodriguez, S. Vadera, and L. E. Sucar. A probabilistic model for case-based reasoning. *Case-Based Reasoning Research and Development*, 1266: 623–632, 1997.

[46] D. Rossille, J.-F. Laurent, and A. Burgun. Modelling a decision-support system for oncology using rule-based and case-based reasoning methodologies. *International journal of medical informatics*, 74(2-4):299–306, 2005.

[47] S. Russell and P. Norvig. *Artificial intelligence: a modern approach.* 1995.

[48] A. Stahl and T. Roth-Berghofer. Rapid prototyping of cbr applications with the open source tool mycbr. *Advances in Case-Based Reasoning*, pages 615–629, 2008.

[49] S. Tamang and D. Kopec. Improving care at the end of life: An automated case-based reasoning approach. In *Third International Conference on Computer Science and its Applications*, San Diego, 2005.

[50] S. Tamang, D. Kopec, G. Shagas, and K. Levy. Improving end of life care: an information systems approach to reducing medical errors. *Studies in health technology and informatics*, 114:93, 2005.

[51] H. Tirri, P. Kontkanen, and P. Myllymaki. A bayesian framework for case-based reasoning. *Advances in Case-Based Reasoning*, 1168:413–427, 1996.

[52] H. M. Tran and J. Schönwälder. Fault representation in case-based reasoning. *LECTURE NOTES IN COMPUTER SCIENCE*, 4785:50, 2007.

[53] I. University of California. Uci machine learning repository, 2009. `http://archive.ics.uci.edu/ml/`.

[54] Vinmonopolet. Vinmonopolet, 2010. `http://www.vinmonopolet.no/`.

[55] Vinmonopolet. Nyttig om mat og vin, 2010.

[56] J. Wyatt and D. Spiegelhalter. Field trials of medical decision-aids: potential problems and solutions. *Proc Annu Symp Comput Appl Med Care*, pages 3–7, 1991.

# Appendices

# Appendix A

# jColibri Similarity Functions

## Ontology Similarity Functions

All four functions are taken from Recio-Garcia et al. [44].

**OntDeepBasic:**

$$\text{fdeep\_basic}(i_1, i_2) = \frac{\max(\text{prof}(\text{LCS}(i_1, i_2)))}{\max\limits_{C_i \in CN} (\text{prof}(C_i))} \tag{A.1}$$

**OntDeep:**

$$\text{fdeep}(i_1, i_2) = \frac{\max(\text{prof}(\text{LCS}(i_1, i_2)))}{\max(\text{prof}(i_1), \text{prof}(i_2))} \tag{A.2}$$

**OntCosine:**

$$\begin{aligned} \text{cosine}(i_1, i_2) &= \\ \text{sim}(t(i_1), t(i_2)) &= \\ &\frac{\left|\left(\cup_{d_i \in t(i_1)}(\text{super}(d_i, CN))\right) \cap \left(\cup_{d_i \in t(i_2)}(\text{super}(d_i, CN)\right)\right|}{\sqrt{\left|\cup_{d_i \in t(i_1)}(\text{super}(d_i, CN))\right|} * \sqrt{\left|\cup_{d_i \in t(i_2)}(\text{super}(d_i, CN)\right|}} \end{aligned} \tag{A.3}$$

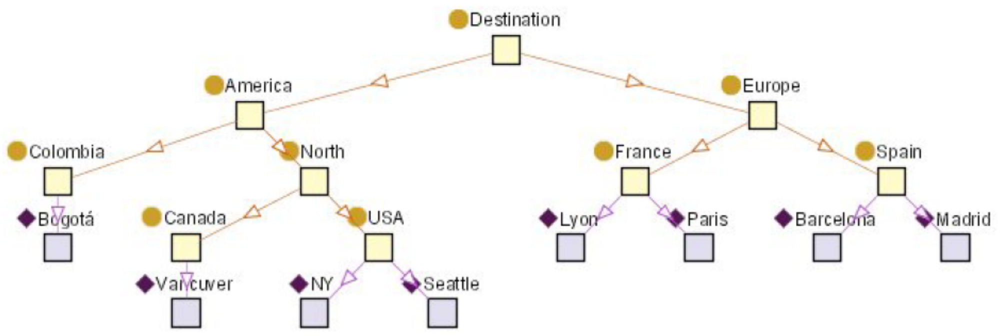
**OntDetail:**

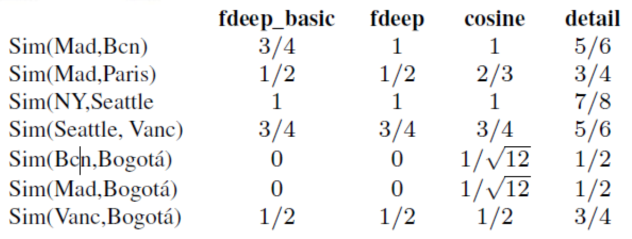
$$\begin{aligned}&\text{detail}(i_1, i_2) = \\ &\text{detail}(t(i_1), t(i_2)) = \\ &1 - \frac{1}{2 * \left|\left(\cup_{d_i \in t(i_1)}(\text{super}(d_i, CN))\right) \cap \left(\cup_{d_i \in t(i_2)}(\text{super}(d_i, CN))\right)\right|}\end{aligned} \tag{A.4}$$

**Where:**

- $CN$ is the set of all concepts in the current knowledge base
- super$(c, C)$ is the subset of concepts in $C$ witch are super concepts of $c$
- LCS$(i_1, i_2)$ is the set of the least common subsumer concepts of the two given individuals
- prof$(c)$ is the depth of concept $c$
- t$(i)$ is the set of concepts the individual $i$ is instance of

These similarity functions measures the degree of similarity between two concepts in the ontology. Figure A.1 illustrates the difference between the ontology similarity functions.





| | fdeep_basic | fdeep | cosine | detail |
|---|---|---|---|---|
| Sim(Mad,Bcn) | 3/4 | 1 | 1 | 5/6 |
| Sim(Mad,Paris) | 1/2 | 1/2 | 2/3 | 3/4 |
| Sim(NY,Seattle | 1 | 1 | 1 | 7/8 |
| Sim(Seattle, Vanc) | 3/4 | 3/4 | 3/4 | 5/6 |
| Sim(Bcn,Bogotá) | 0 | 0 | $1/\sqrt{12}$ | 1/2 |
| Sim(Mad,Bogotá) | 0 | 0 | $1/\sqrt{12}$ | 1/2 |
| Sim(Vanc,Bogotá) | 1/2 | 1/2 | 1/2 | 3/4 |

Figure A.1: Example of how the similarity functions works. From Recio-Garcia et al. [44].

## Equal

Listing A.1: The compute method in the Equal() class in jColibri.

```
public double compute(Object o1, Object o2)
throws NoApplicableSimilarityFunctionException{
        if ((o1 == null) || (o2 == null)){
                return 0;
  return o1.equals(o2) ? 1 : 0;
}
```

The Equal function compares two objects (i.e., two text strings) and returns 1 if there is a match and 0 if not. In the last line, the function redirects the call to the equals method of the Object class, which returns true if the two objects are the same, or false if they are not.

# Appendix B

# Wine terms

## Grape mixes

This section will shortly describe the grape mixes mentioned in the thesis.

**Bardolino** An Italian wine from Verona. It is a blend of Corvina, Rondinella and Molinara.

**Bordeaux** Any wine produced in the Bordeaux region in France. The most used grapes are Cabernet Sauvignon, Cabernet Franc, Merlot and Petit Verdot.

**Chianti** The most famous red wine from Italy. Has to have at least 80% Sangiovese, the rest is up to the farmer.

**Valpolicella** An Italian wine from Veneto in North-Italy. Is a mix of Corvina Veronese (40-70%), Rondinella (20-40%) and Molinara (5-20%).

**Tinta Roriz** This is not a mixture, but the Portugese name for Tempranillo. All wines are made of Tinta Roriz is therefore classified as Tempranillo wines.

# Appendix C

# Test tables

## Test of the second BN

Because of unsatisfying results from the first BN, we revised the BN as elaborated in section 5.2.2. The results from the tests are described in section 5.2.

| **Grapes** | **Without Evidence** |
|---|---|
| Barbera | 7.1 % |
| Cabernet Franc | 6.0 % |
| Cabernet Sauvignon | 20.8 % |
| Carignan | 3.4 % |
| Corvina | 4.8 % |
| Grenache | 7.5 % |
| Malbec | 1.7 % |
| Merlot | 10.0 % |
| Montepulciano | 2.1 % |
| Pinot Noir | 6.1 % |
| Rondinella | 3.7 % |
| Sangiovese | 6.0 % |
| Syrah | 9.9 % |
| Tempranillo | 7.7 % |
| Tinta Miuda | 0.3 % |
| Zinfandel | 0.5 % |

Table C.1: Results from the revised BN without evidence. This results were used to calculate the log-odds ratio for the different grapes.

| Course 1 - Chicken | | | |
|---|---|---|---|
| **Rank** | **Grapes** | **With Evidence** | **Log-Odds Ratio** |
| 1 | Zinfandel | 0.59 % | 0.175 |
| 2 | Montepulciano | 2.49 % | 0.154 |
| 3 | Tempranillo | 8.23 % | 0.065 |
| 4 | Barbera | 7.55 % | 0.059 |
| 5 | PinotNoir | 6.35 % | 0.04 |
| 6 | Rondinella | 3.8 % | 0.022 |
| 7 | TintaMiuda | 0.31 % | -0.001 |
| 8 | Corvina | 4.81 % | -0.006 |
| 9 | CabernetSauvignon | 19.88 % | -0.047 |
| 10 | Carignan | 3.17 % | -0.056 |
| 11 | Grenache | 6.83 % | -0.092 |
| 12 | Merlot | 9.12 % | -0.096 |
| 13 | Sangiovese | 5.38 % | -0.105 |
| 14 | Syrah | 8.51 % | -0.147 |
| 15 | CabernetFranc | 4.93 % | -0.191 |
| 16 | Malbec | 1.29 % | -0.264 |
| Matching suggestions | | 1 | |

Table C.2: BN Results from second test. Vinmonopolet recommends Pinot Noir, Barbera and Cabernet Franc for this meal.

| Course 2 - Coq | | | |
|---|---|---|---|
| **Rank** | **Grapes** | **With Evidence** | **Log-Odds Ratio** |
| 1 | Barbera | 8.2 % | 0.136 |
| 2 | Cabernet Sauvignon | 22.4 % | 0.071 |
| 3 | Carignan | 3.6 % | 0.060 |
| 4 | Cabernet Franc | 6.3 % | 0.052 |
| 5 | Grenache | 7.9 % | 0.049 |
| 6 | Malbec | 1.7 % | -0.017 |
| 7 | Tempranillo | 7.6 % | -0.022 |
| 8 | Merlot | 9.8 % | -0.025 |
| 9 | Syrah | 9.5 % | -0.034 |
| 10 | Corvina | 4.6 % | -0.055 |
| 11 | Sangiovese | 5.6 % | -0.073 |
| 12 | Tinta Miuda | 0.3 % | -0.076 |
| 13 | Montepulciano | 2.0 % | -0.089 |
| 14 | Zinfandel | 0.4 % | -0.094 |
| 15 | Pinot Noir | 5.5 % | -0.025 |
| 16 | Rondinella | 3.2 % | -0.149 |
| Matching suggestions | | 0 | |

Table C.3: BN Results from second test. Vinmonopolet recommends Pinot Noir, Syrah and Cabernet Franc for this meal.

| **Course 3 - Beef** | | | |
|---|---|---|---|
| **Rank** | **Grapes** | **With Evidence** | **Log-Odds Ratio** |
| 1 | CabernetFranc | 7.1 % | 0.173 |
| 2 | Malbec | 1.9 % | 0.124 |
| 3 | Syrah | 10.7 % | 0.08 |
| 4 | Grenache | 8.1 % | 0.075 |
| 5 | Zinfandel | 0.5 % | 0.074 |
| 6 | Barbera | 7.4 % | 0.044 |
| 7 | Merlot | 10.2 % | 0.017 |
| 8 | CabernetSauvignon | 21.2 % | 0.017 |
| 9 | Carignan | 3.4 % | 0.009 |
| 10 | Tempranillo | 7.7 % | -0.001 |
| 11 | Sangiovese | 5.9 % | -0.015 |
| 12 | PinotNoir | 5.5 % | -0.108 |
| 13 | Corvina | 4.3 % | -0.109 |
| 14 | Rondinella | 3.1 % | -0.192 |
| 15 | Montepulciano | 1.5 % | -0.377 |
| 16 | TintaMiuda | 0.1 % | -0.818 |
| Matching suggestions | | 1 | |

Table C.4: BN Results from second test. Vinmonopolet recommends Chianti (Sangiovese) and Bordeaux (Cabernet Sauvignon, Cabernet Franc, Merlot and Petit Verdot) for this meal.

| **Course 4 - Pork fillet** | | | |
|---|---|---|---|
| **Rank** | **Grapes** | **With Evidence** | **Log-Odds Ratio** |
| 1 | TintaMiuda | 0.4 % | 0.369 |
| 2 | Montepulciano | 2.6 % | 0.181 |
| 3 | Barbera | 8.3 % | 0.155 |
| 4 | CabernetSauvignon | 23.9 % | 0.137 |
| 5 | Carignan | 3.6 % | 0.084 |
| 6 | Corvina | 5 % | 0.028 |
| 7 | Grenache | 7.5 % | 0.006 |
| 8 | Tempranillo | 7.6 % | -0.018 |
| 9 | Merlot | 9.7 % | -0.037 |
| 10 | CabernetFranc | 5.7 % | -0.043 |
| 11 | Rondinella | 3.5 % | -0.065 |
| 12 | PinotNoir | 5.6 % | -0.085 |
| 13 | Sangiovese | 5.5 % | -0.09 |
| 14 | Malbec | 1.5 % | -0.098 |
| 15 | Syrah | 8.4 % | -0.16 |
| 16 | Zinfandel | 0.4 % | -0.257 |
| Matching suggestions | | 0 | |

Table C.5: BN Results from second test. Vinmonopolet recommends Cabernet Franc and Pinot Noir for this meal.

| **Course 5 - Pork rib** | | | |
|---|---|---|---|
| **Rank** | **Grapes** | **With Evidence** | **Log-Odds Ratio** |
| 1 | Barbera | 8.5 % | 0.181 |
| 2 | CabernetSauvignon | 23.5 % | 0.119 |
| 3 | TintaMiuda | 0.3 % | 0.089 |
| 4 | Carignan | 3.7 % | 0.087 |
| 5 | CabernetFranc | 6.5 % | 0.086 |
| 6 | Grenache | 8 % | 0.066 |
| 7 | Malbec | 1.7 % | 0.013 |
| 8 | Tempranillo | 7.7 % | -0.007 |
| 9 | Merlot | 9.8 % | -0.024 |
| 10 | Corvina | 4.5 % | -0.076 |
| 11 | Syrah | 9.1 % | -0.08 |
| 12 | Sangiovese | 5.5 % | -0.086 |
| 13 | Montepulciano | 1.9 % | -0.103 |
| 14 | PinotNoir | 5.2 % | -0.151 |
| 15 | Zinfandel | 0.4 % | -0.17 |
| 16 | Rondinella | 3 % | -0.219 |
| Matching suggestions | | 0 | |

Table C.6: BN Results from second test. Vinmonopolet recommends Sangiovese and Pinot Noir for this meal.

| **Course 6 - Lamb roast** | | | |
|---|---|---|---|
| **Rank** | **Grapes** | **With Evidence** | **Log-Odds Ratio** |
| 1 | Malbec | 2.2 % | 0.247 |
| 2 | CabernetFranc | 7.1 % | 0.173 |
| 3 | Grenache | 8.7 % | 0.153 |
| 4 | Syrah | 11.3 % | 0.137 |
| 5 | Zinfandel | 0.5 % | 0.096 |
| 6 | Barbera | 7.8 % | 0.091 |
| 7 | Sangiovese | 6.4 % | 0.065 |
| 8 | Carignan | 3.6 % | 0.064 |
| 9 | Merlot | 10.3 % | 0.024 |
| 10 | CabernetSauvignon | 19.6 % | -0.062 |
| 11 | Tempranillo | 7.1 % | -0.085 |
| 12 | Corvina | 4.3 % | -0.114 |
| 13 | PinotNoir | 5.3 % | -0.144 |
| 14 | Rondinella | 3 % | -0.207 |
| 15 | Montepulciano | 1.4 % | -0.416 |
| 16 | TintaMiuda | 0.1 % | -0.894 |
| Matching suggestions | | 0 | |

Table C.7: BN Results from second test. Vinmonopolet recommends Chianti (Sangiovese) and Tempranillo for this meal.

| **Course 7 - ”Fårikål”** | | | |
|---|---|---|---|
| **Rank** | **Grapes** | **With Evidence** | **Log-Odds Ratio** |
| 1 | Malbec | 1.9 % | 0.138 |
| 2 | Grenache | 8.5 % | 0.129 |
| 3 | Barbera | 8.1 % | 0.123 |
| 4 | Syrah | 10.8 % | 0.095 |
| 5 | Carignan | 3.6 % | 0.073 |
| 6 | CabernetFranc | 6.4 % | 0.068 |
| 7 | Zinfandel | 0.5 % | 0.042 |
| 8 | Merlot | 10.3 % | 0.02 |
| 9 | Sangiovese | 6.1 % | 0.019 |
| 10 | PinotNoir | 5.9 % | -0.029 |
| 11 | CabernetSauvignon | 19.7 % | -0.055 |
| 12 | Corvina | 4.5 % | -0.077 |
| 13 | Tempranillo | 7 % | -0.095 |
| 14 | Rondinella | 3.2 % | -0.148 |
| 15 | Montepulciano | 1.8 % | -0.181 |
| 16 | TintaMiuda | 0.2 % | -0.474 |
| Matching suggestions | | 0 | |

Table C.8: BN Results from second test. Vinmonopolet recommends Valpolicella and Bardolino for this meal. Both of these are grape mixes consisting of Corvina, Rondinella and Molinara.

| **Course 8 - Bacalao** | | | |
|---|---|---|---|
| **Rank** | **Grapes** | **With Evidence** | **Log-Odds Ratio** |
| 1 | Tinta Miuda | 0.6 % | 0.600 |
| 2 | Montepulciano | 3.6 % | 0.521 |
| 3 | Rondinella | 4.9 % | 0.282 |
| 4 | Corvina | 6.1 % | 0.225 |
| 5 | Pinot Noir | 6.9 % | 0.121 |
| 6 | Cabernet Sauvignon | 21.3 % | 0.021 |
| 7 | Merlot | 9.7 % | -0.034 |
| 8 | Carignan | 3.2 % | -0.033 |
| 9 | Sangiovese | 5.8 % | -0.035 |
| 10 | Tempranillo | 7.3 % | -0.056 |
| 11 | Barbera | 6.7 % | -0.064 |
| 12 | Syrah | 8.3 % | -0.169 |
| 13 | Grenache | 6.3 % | -0.175 |
| 14 | Zinfandel | 0.4 % | -0.211 |
| 15 | Cabernet Franc | 4.4 % | -0316 |
| 16 | Malbec | 1.2 % | -0.378 |
| Matching suggestions | | 2 | |

Table C.9: BN Results from second test. Vinmonopolet recommends Barbera, Valpolicella (mix of Corvina, Rondinella and Molinara) and Portuguese wines for this meal.

| **Course 9 - Halibut** | | | |
|---|---|---|---|
| **Rank** | **Grapes** | **With Evidence** | **Log-Odds Ratio** |
| 1 | Montepulciano | 3.7 % | 0.559 |
| 2 | TintaMiuda | 0.5 % | 0.534 |
| 3 | Rondinella | 5.1 % | 0.309 |
| 4 | Corvina | 6.3 % | 0.258 |
| 5 | PinotNoir | 7.2 % | 0.17 |
| 6 | CabernetSauvignon | 20.8 % | -0.004 |
| 7 | Merlot | 9.9 % | -0.014 |
| 8 | Carignan | 3.2 % | -0.04 |
| 9 | Sangiovese | 5.7 % | -0.042 |
| 10 | Tempranillo | 7.4 % | -0.044 |
| 11 | Barbera | 6.7 % | -0.055 |
| 12 | Syrah | 8.1 % | -0.196 |
| 13 | Zinfandel | 0.4 % | -0.197 |
| 14 | Grenache | 6.1 % | -0.201 |
| 15 | CabernetFranc | 4.3 % | -0.331 |
| 16 | Malbec | 1.1 % | -0.412 |
| Matching suggestions | | 1 | |

Table C.10: BN Results from second test. Vinmonopolet recommends Valpolicella (mix of Corvina, Rondinella and Molinara) and Pinot Noir for this meal.

| **Course 10 - Gorgonzola** | | | |
|---|---|---|---|
| **Rank** | **Grapes** | **With Evidence** | **Log-Odds Ratio** |
| 1 | TintaMiuda | 0.4 % | 0.247 |
| 2 | CabernetSauvignon | 22 % | 0.054 |
| 3 | Montepulciano | 2.2 % | 0.034 |
| 4 | Tempranillo | 7.9 % | 0.027 |
| 5 | Merlot | 10.3 % | 0.026 |
| 6 | PinotNoir | 6.2 % | 0.01 |
| 7 | Carignan | 3.4 % | 0.009 |
| 8 | Corvina | 4.9 % | 0.005 |
| 9 | Rondinella | 3.7 % | -0.004 |
| 10 | Sangiovese | 5.9 % | -0.007 |
| 11 | Syrah | 9.8 % | -0.008 |
| 12 | CabernetFranc | 5.8 % | -0.034 |
| 13 | Zinfandel | 0.5 % | -0.036 |
| 14 | Barbera | 6.9 % | -0.038 |
| 15 | Grenache | 7.2 % | -0.046 |
| 16 | Malbec | 1.6 % | -0.074 |
| Matching suggestions | | 0 | |

Table C.11: BN Results from second test. Vinmonopolet recommends Valpolicella (mix of Corvina, Rondinella and Molinara) and Shiraz (an other name for Syrah) for this meal.

| **Course 11 - Hare roast** | | | |
|---|---|---|---|
| **Rank** | **Grapes** | **With Evidence** | **Log-Odds Ratio** |
| 1 | Malbec | 2.1 % | 0.227 |
| 2 | Zinfandel | 0.6 % | 0.199 |
| 3 | Syrah | 11.8 % | 0.183 |
| 4 | Sangiovese | 6.6 % | 0.098 |
| 5 | Grenache | 8 % | 0.07 |
| 6 | CabernetFranc | 6.3 % | 0.047 |
| 7 | Merlot | 10.5 % | 0.045 |
| 8 | Carignan | 3.2 % | -0.039 |
| 9 | PinotNoir | 5.7 % | -0.063 |
| 10 | Corvina | 4.5 % | -0.073 |
| 11 | Rondinella | 3.5 % | -0.074 |
| 12 | Barbera | 6.1 % | -0.149 |
| 13 | CabernetSauvignon | 17.8 % | -0.16 |
| 14 | Tempranillo | 6.5 % | -0.178 |
| 15 | Montepulciano | 1.5 % | -0.343 |
| 16 | TintaMiuda | 0.1 % | -1.137 |
| Matching suggestions | | 2 | |

Table C.12: BN Results from second test. Vinmonopolet recommends Syrah and Zinfandel for this meal.

| **Course 12 - Elk roast** | | | |
|---|---|---|---|
| **Rank** | **Grapes** | **With Evidence** | **Log-Odds Ratio** |
| 1 | Malbec | 2.2 % | 0.252 |
| 2 | CabernetFranc | 7.2 % | 0.192 |
| 3 | Zinfandel | 0.6 % | 0.164 |
| 4 | Syrah | 11.5 % | 0.155 |
| 5 | Sangiovese | 6.7 % | 0.107 |
| 6 | Grenache | 8.2 % | 0.094 |
| 7 | Merlot | 10.5 % | 0.042 |
| 8 | Carignan | 3.4 % | 0.006 |
| 9 | CabernetSauvignon | 20.9 % | 0.002 |
| 10 | Barbera | 6.7 % | -0.057 |
| 11 | Corvina | 4.6 % | -0.059 |
| 12 | Tempranillo | 7.2 % | -0.074 |
| 13 | Rondinella | 3.4 % | -0.097 |
| 14 | PinotNoir | 5.2 % | -0.155 |
| 15 | Montepulciano | 1.5 % | -0.378 |
| 16 | TintaMiuda | 0.2 % | -0.63 |
| Matching suggestions | | 0 | |

Table C.13: BN Results from second test. Vinmonopolet recommends Syrah and Sangiovese for this meal.

| **Course 13 - Salad** | | | |
|---|---|---|---|
| **Rank** | **Grapes** | **With Evidence** | **Log-Odds Ratio** |
| 1 | Tinta Miuda | 0.7 % | 0.863 |
| 2 | Rondinella | 6.6 % | 0.572 |
| 3 | Pinot Noir | 10.6 % | 0.553 |
| 4 | Montepulciano | 3.3 % | 0.43 |
| 5 | Tempranillo | 10.9 % | 0.346 |
| 6 | Corvina | 5.6 % | 0.151 |
| 7 | Sangiovese | 5.5 % | -0.085 |
| 8 | Merlot | 8.7 % | -0.148 |
| 9 | Cabernet Sauvignon | 16.6 % | -0.225 |
| 10 | Carignan | 2.7 % | -0.236 |
| 11 | Zinfandel | 0.4 % | -0.313 |
| 12 | Barbera | 4.9 % | -0.383 |
| 13 | Syrah | 6.7 % | -0.384 |
| 14 | Grenache | 5.0 % | -0.406 |
| 15 | Cabernet Franc | 3.0 % | -0.699 |
| 16 | Malbec | 0.7 % | -0.851 |
| Matching suggestions | | 0 | |

Table C.14: BN Results from second test. Vinmonopolet recommends Cabernet Franc and Barbera for this meal.

## Test of the system as a whole

In order to see if the system could have any actual benefit, we tested the whole system. The results were evaluated by an employee at Vinmonopolet Solsiden.

| **Course 1 - Chicken, oven-baked with root vegetables** | | | |
|---|---|---|---|
| **#** | **Similarity** | **Wine Suggestion** | **Used in course** |
| 1 | 0.79 | Ravenswood Zen of Zin 2004 | Grilled chicken-legs |
| 2 | 0.73 | Marques de la Concordia Reserva 2005 | Soft cheeses |
| 3 | 0.72 | Vidigal Reserva 2005 | Coq au wine |
| 4 | 0.45 | Zaccagnini Montepulciano d'Abruzzo 2005 | Pheasantbreast with olive-gravy |
| 5 | 0.45 | Condado de Haza Crianza 2006 | Hare fillet with pepper sauce |

(a) The case results.

| # | Meal contents |
|---|---|
| 1 | ChickenFillet, Salad, NoSauce |
| 2 | CheeseDish, Vegetable, NoSauce |
| 3 | Cock, Rice, Mushroom, Vegetable, Bread, BoiledPotatoes, GravySauce |
| 4 | Pheasant, Vegetable, GravySauce |
| 5 | Hare, Vegetable, GameSauce |

(b) The contents of the results.

| # | Grapes |
|---|---|
| 1 | Zinfandel |
| 2 | Tempranillo |
| 3 | Tempranillo, TintaMiuda |
| 4 | Montepulciano |
| 5 | Tempranillo |

(c) The grapes present in the wines.

Table C.15: Test results course 1.

The results in table C.15 was unfortunately not evaluated because we had an incorrect table at the day of the testing. We will individually evaluate the results based on information found about the different wines at `http://www.aperitif.no/`. Ravenswood Zen of Zin 2004 is a heavy wine, and is said to be used with grilled and fried meat that is juicy. It also goes well to oven baked or braised meat.

We therefore judge this wine as adequate. The next wine, Marques de Concordia Reserva 2005, has a round fruitiness and is recommended to drink together with lamb and cheese. According to `http://www.vinmonopolet.no/`, this wine also goes to light meat and beef. This wine could therefore also be adequate for this course. Vidigal Reserva 2005 can be used together with lamb, light meat, pasta and pizza. It is also said by Apéritif to be juicy, which Hoøen said was an important characteristic for this course. Zaccagnini Montepulciano d'Abruzzo 2005 is another juicy wine. This wine can be used together with tasty pasta dishes, oven baked meat and all round read meat. Since this wine goes to oven baked meat, it might also go well to this course.The last wine, Condado de Haza Crianza 2006, is a rich wine that goes to red meat and game. This wine does not seem to be an adequate match for our meal.

| Course 2 - Coq au wine | | | |
|---|---|---|---|
| **#** | **Similarity** | **Wine Suggestion** | **Used in course** |
| 1 | 0.73 | Isole e Olena Chianti Classico 2004 | Elk stew |
| 2 | 0.73 | Gran Coronas Reserva 2005 | Stewed beef |
| 3 | 0.59 | Wolf Blass Yellow Label Cabernet Sauvignon 2003/2004 | Elk steak |
| 4 | 0.59 | Villa Antinori 2005 | Baked calf fillet |
| 5 | 0.57 | Carmen Cabernet Sauvignon 2004 | Roedeer fillet |

(a) The case results.

| # | **Meal contents** |
|---|---|
| 1 | Elk, Carrots, BoiledPotatoes, Vegetable, Onion, GravySauce |
| 2 | BeefSirloin, BoiledPotatoes, Vegetable, GravySauce |
| 3 | Elk, BoiledPotatoes, Asparges, Mushroom, GravySauce |
| 4 | Calf, Vegetable, MashedPotatoes, RedWineSauce |
| 5 | Roedeer, Pasta, Mushroom, GravySauce |

(b) The contents of the results.

| # | **Grapes** |
|---|---|
| 1 | CabernetSauvignon, Sangiovese, Syrah |
| 2 | CabernetSauvignon, Tempranillo |
| 3 | CabernetSauvignon |
| 4 | CabernetSauvignon, Sangiovese, Syrah, Merlot |
| 5 | CabernetSauvignon |

(c) The grapes present in the wines.

Table C.16: Test results course 2

| Course 3 - Beef with baked potatoes and Bearnaise | | | |
|---|---|---|---|
| **#** | **Similarity** | **Wine Suggestion** | **Used in course** |
| 1 | 0.68 | Domaine de l'Ameillaud Cairanne 2006 | Beef with asparges, mushrooms and potato puree |
| 2 | 0.51 | Elsa Malbec 2005 | Beef Pebre |
| 3 | 0.45 | Vidal-Fleury Crozes-Hermitage 2006 | Reindeer fillet |
| 4 | 0.43 | Ch. Franc Cardinal 2002 | Beef with aroma butter and potato salad |
| 5 | 0.42 | Lindemans Cawarra Shiraz Cabernet 2007 | Reindeer stew with cheese sauce |

(a) The case results.

| # | Meal contents |
|---|---|
| 1 | BeefTenderloin, Mushroom, Asparges, MashedPotatoes, CreamSauce |
| 2 | BeefTenderloin, Mushroom, Rice, NoSauce |
| 3 | Reindeer, Mushroom, BakedPotatoes, GameSauce |
| 4 | BeefSirloin, Salad, BoiledPotatoes, NoSauce |
| 5 | Reindeer, Mushroom, BoiledPotatoes, Vegetable, CreamSauce |

(b) The contents of the results.

| # | Grapes |
|---|---|
| 1 | Carignan, Syrah, Grenache |
| 2 | Malbec |
| 3 | Syrah |
| 4 | Merlot, CabernetFranc |
| 5 | Syrah, CabernetSauvignon |

(c) The grapes present in the wines.

Table C.17: Test results course 3

| **Course 4 - Pork, fillet** | | | |
|---|---|---|---|
| **#** | **Similarity** | **Wine Suggestion** | **Used in course** |
| 1 | 1 | La Trincherina Barbera d'Asti 2007 | Exotic pork fillet |
| 2 | 0.45 | Vidigal Reserva 2005 | Coq au wine |
| 3 | 0.45 | Zaccagnini Montepulciano d'Abruzzo 2005 | Pheasantbreast with olive-gravy |
| 4 | 0.45 | Torriglione Barbera d'Alba 2004/2005 | Turkey |
| 5 | 0.24 | Vidigal Reserva 2005 | Chilicannelloni |

(a) The case results.

| # | **Meal contents** |
|---|---|
| 1 | PorkSirloin, Garlic, Vegetable, NoSauce |
| 2 | Cock, BoiledPotatoes, Mushroom, Bread, Vegetable, Rice, GravySauce |
| 3 | Pheasant, Vegetable, GravySauce |
| 4 | WholeTurkey, BoiledPotatoes, Vegetable, GravySauce |
| 5 | VegetarianDish, Bread, Salad, TomatoSauce |

(b) The contents of the results.

| # | **Grapes** |
|---|---|
| 1 | Barbera |
| 2 | Tempranillo, TintaMiuda |
| 3 | Montepulciano |
| 4 | Barbera |
| 5 | Tempranillo, Malbec, TintaMiuda |

(c) The grapes present in the wines.

Table C.18: Test results course 4

| Course 5 - Pork rib | | | |
|---|---|---|---|
| **#** | **Similarity** | **Wine Suggestion** | **Used in course** |
| 1 | 0.73 | Vidigal Reserva 2005 | Coq au wine |
| 2 | 0.73 | Wolf Blass Yellow Label Cabernet Sauvignon 2003/2004 | Elk steak |
| 3 | 0.73 | Isole e Olena Chianti Classico 2004 | Elk stew |
| 4 | 0.73 | Gran Coronas Reserva 2005 | Stewed beef |
| 5 | 0.52 | Carmen Cabernet Sauvignon 2004 | Roedeer fillet |

(a) The case results.

| # | Meal contents |
|---|---|
| 1 | Cock, Bread, Rice, BoiledPotatoes, Mushroom, Vegetable, GravySauce |
| 2 | Elk, Asparges, BoiledPotatoes, Mushroom, GravySauce |
| 3 | Elk, Carrots, Onion, BoiledPotatoes, Vegetable, GravySauce |
| 4 | BeefSirloin, BoiledPotatoes, Vegetable, GravySauce |
| 5 | Roedeer, Pasta, Mushroom, GravySauce |

(b) The contents of the results.

| # | Grapes |
|---|---|
| 1 | TintaMiuda, Tempranillo |
| 2 | CabernetSauvignon |
| 3 | Sangiovese, CabernetSauvignon, Syrah |
| 4 | Tempranillo, CabernetSauvignon |
| 5 | CabernetSauvignon |

(c) The grapes present in the wines.

Table C.19: Test results course 5

| Course 6 - Lamb, roast | | | |
|---|---|---|---|
| **#** | **Similarity** | **Wine Suggestion** | **Used in course** |
| 1 | 1 | Domaine d'Andézon 2007 | Lamb roast |
| 2 | 0.87 | Amirault La Coudraye 2007 | Lamb cutlets with thyme and garlic potato purée |
| 3 | 0.71 | Arne Brimis Viltvin 2007 | Mouton á la Bourguignonne |
| 4 | 0.57 | Guigal Côtes du Rhône 2005 | Steamed spring cod |
| 5 | 0.45 | Domaine de l'Ameillaud Cairanne 2006 | Beef with asparges, mushrooms and potato puree |

(a) The case results.

| # | Meal contents |
|---|---|
| 1 | LambRoast, Vegetable, MashedPotatoes, RedWineSauce |
| 2 | LambCutlets, MashedPotatoes, RedWineSauce |
| 3 | LambStew, BoiledPotatoes, Rice, RedWineSauce |
| 4 | Cod, Tomatoes, BoiledPotatoes, RedWineSauce |
| 5 | BeefTenderloin, Asparges, Mushroom, MashedPotatoes, CreamSauce |

(b) The contents of the results.

| # | Grapes |
|---|---|
| 1 | Grenache, Syrah |
| 2 | CabernetFranc |
| 3 | Grenache, Syrah |
| 4 | Grenache, Syrah |
| 5 | Carignan, Grenache, Syrah |

(c) The grapes present in the wines.

Table C.20: Test results course 6

| Course 7 - "Fårikål" | | | |
|---|---|---|---|
| **#** | **Similarity** | **Wine Suggestion** | **Used in course** |
| 1 | 1 | Domaine Robert Vic Cuvée Sélection 2008 | Norwegian lamb stew |
| 2 | 0.63 | Oléa Signature 2008 | Steamed cod on a bed of lentils with vegetables |
| 3 | 0.63 | La Trincherina Barbera d'Asti 2007 | Exotic pork fillet |
| 4 | 0.57 | Elsa Malbec 2005 | Beef Pebre |
| 5 | 0.52 | Argento Malbec 2008/2009 | Cheese table |

(a) The case results.

| # | Meal contents |
|---|---|
| 1 | Mutton, BoiledPotatoes, Cabbage, NoSauce |
| 2 | Cod, Vegetable, NoSauce |
| 3 | PorkSirloin, Vegetable, Garlic, NoSauce |
| 4 | BeefTenderloin, Mushroom, Rice, NoSauce |
| 5 | CheeseDish, Salad, NoSauce |

(b) The contents of the results.

| # | Grapes |
|---|---|
| 1 | Syrah, Grenache, Merlot |
| 2 | Carignan, Grenache |
| 3 | Barbera |
| 4 | Malbec |
| 5 | Malbec |

(c) The grapes present in the wines.

Table C.21: Test results course 7

| Course 8 - Bacalao | | | |
|---|---|---|---|
| **#** | **Similarity** | **Wine Suggestion** | **Used in course** |
| 1 | 0.52 | Vidigal Reserva 2005 | Chilicannelloni |
| 2 | 0.45 | Vidigal Reserva 2005 | Coq au wine |
| 3 | 0.45 | Zenato Valpolicella Classico Superiore 2007 | Pork rib |
| 4 | 0.37 | Duca del Frassino Valpolicella Superiore 2007 | Baked coalfish |
| 5 | 0.32 | Zenato Valpolicella Classico Superiore 2007 | Turkey fillet on a bed of vegetables |

(a) The case results.

| # | Meal contents |
|---|---|
| 1 | VegetarianDish, Bread, Salad, TomatoSauce |
| 2 | Cock, BoiledPotatoes, Rice, Mushroom, Bread, Vegetable, GravySauce |
| 3 | PorkRib, BoiledPotatoes, Vegetable, GravySauce |
| 4 | Coalfish, Onion, Tomatoes, Pepper, Garlic, NoSauce |
| 5 | TurkeyFillet, Vegetable, RedWineSauce |

(b) The contents of the results.

| # | Grapes |
|---|---|
| 1 | Tempranillo, Malbec, TintaMiuda |
| 2 | Tempranillo, TintaMiuda |
| 3 | Rondinella, Sangiovese, Corvina, Molinara |
| 4 | Rondinella, Corvina, Molinara |
| 5 | Rondinella, Sangiovese, Corvina, Molinara |

(c) The grapes present in the wines.

Table C.22: Test results course 8

| **Course 9 - Halibut, fried/grilled** | | | |
|---|---|---|---|
| **#** | **Similarity** | **Wine Suggestion** | **Used in course** |
| 1 | 0.48 | Duca del Frassino Valpolicella Superiore 2007 | Baked coalfish |
| 2 | 0.45 | Vidigal Reserva 2005 | Coq au wine |
| 3 | 0.45 | Zaccagnini Montepulciano d'Abruzzo 2005 | Pheasantbreast with olive-gravy |
| 4 | 0.45 | Zenato Valpolicella Classico Superiore 2007 | Turkey fillet on a bed of vegetables |
| 5 | 0.45 | Zenato Valpolicella Classico Superiore 2007 | Pork rib |

(a) The case results.

| # | **Meal contents** |
|---|---|
| 1 | Coalfish, Pepper, Tomatoes, Garlic, Onion, NoSauce |
| 2 | Cock, Bread, Vegetable, BoiledPotatoes, Mushroom, Rice, GravySauce |
| 3 | Pheasant, Vegetable, GravySauce |
| 4 | TurkeyFillet, Vegetable, RedWineSauce |
| 5 | PorkRib, Vegetable, BoiledPotatoes, GravySauce |

(b) The contents of the results.

| # | **Grapes** |
|---|---|
| 1 | Rondinella, Corvina, Molinara |
| 2 | Tempranillo, TintaMiuda |
| 3 | Montepulciano |
| 4 | Rondinella, Corvina, Molinara, Sangiovese |
| 5 | Rondinella, Corvina, Molinara, Sangiovese |

(c) The grapes present in the wines.

Table C.23: Test results course 9

| Course 10 - Gorgonzola | | | |
|---|---|---|---|
| **#** | **Similarity** | **Wine Suggestion** | **Used in course** |
| 1 | 0.41 | Canepa Estate Cabernet Sauvignon 2005 | Lamb cutlets with champignons and feta cheese |
| 2 | 0.41 | La Roche 2004 | Gamepan |
| 3 | 0.41 | I Sodi del Paretaio Chianti 2008 | Chicken fillet with Parma ham and Parmesan cheese |
| 4 | 0.13 | Vidigal Reserva 2005 | Coq au wine |
| 5 | 0.13 | Wolf Blass Yellow Label Cabernet Sauvignon 2003/2004 | Elk steak |

(a) The case results.

| # | **Meal contents** |
|---|---|
| 1 | LambCutlets, Garlic, Onion, Mushroom, NoSauce |
| 2 | Reindeer, MashedPotatoes, Vegetable, NoSauce |
| 3 | ChickenFillet, Pasta, NoSauce |
| 4 | Cock, Rice, Vegetable, BoiledPotatoes, Bread, Mushroom, GravySauce |
| 5 | Elk, Asparges, BoiledPotatoes, Mushroom, GravySauce |

(b) The contents of the results.

| # | **Grapes** |
|---|---|
| 1 | CabernetSauvignon |
| 2 | CabernetSauvignon, Merlot |
| 3 | Syrah, Sangiovese, Merlot, CabernetSauvignon |
| 4 | TintaMiuda, Tempranillo |
| 5 | CabernetSauvignon |

(c) The grapes present in the wines.

Table C.24: Test results course 10

| **Course 11 - Hare, roast** | | | |
|---|---|---|---|
| **#** | **Similarity** | **Wine Suggestion** | **Used in course** |
| 1 | 0.72 | Gaillard Côte-Rôtie 2002 | Grouse breast |
| 2 | 0.59 | Isole e Olena Chianti Classico 2004 | Elk stew |
| 3 | 0.57 | Vidal-Fleury Crozes-Hermitage 2006 | Reindeer fillet |
| 4 | 0.45 | Domaine d'Andézon 2007 | Lamb roast |
| 5 | 0.45 | Villa Antinori 2005 | Baked calf fillet |

(a) The case results.

| # | **Meal contents** |
|---|---|
| 1 | Grouse, Onion, Carrots, BoiledPotatoes, GameSauce |
| 2 | Elk, Carrots, Onion, Vegetable, BoiledPotatoes, GravySauce |
| 3 | Reindeer, BakedPotatoes, Mushroom, GameSauce |
| 4 | LambRoast, Vegetable, MashedPotatoes, RedWineSauce |
| 5 | Calf, Vegetable, MashedPotatoes, RedWineSauce |

(b) The contents of the results.

| # | **Grapes** |
|---|---|
| 1 | Syrah |
| 2 | Sangiovese, CabernetSauvignon, Syrah |
| 3 | Syrah |
| 4 | Grenache, Syrah |
| 5 | Merlot, Sangiovese, CabernetSauvignon, Syrah |

(c) The grapes present in the wines.

Table C.25: Test results course 11

| Course 12 - Elk, roast | | | |
|---|---|---|---|
| **#** | **Similarity** | **Wine Suggestion** | **Used in course** |
| 1 | 0.73 | Beringer Founders' Estate Zinfandel 2004 | Beef and creamed onion |
| 2 | 0.29 | Elsa Malbec 2005 | Beef Pebre |
| 3 | 0.24 | Amirault La Coudraye 2007 | Lamb cutlets with thyme and garlic potato purée |
| 4 | 0.24 | Ravenswood Zen of Zin 2004 | Grilled chicken-legs |
| 5 | 0.24 | Ch. Franc Cardinal 2002 | Beef with aroma butter and potato salad |

(a) The case results.

| # | Meal contents |
|---|---|
| 1 | BeefSirloin, Carrots, Vegetable, Onion, BoiledPotatoes, CreamSauce |
| 2 | BeefTenderloin, Rice, Mushroom, NoSauce |
| 3 | LambCutlets, MashedPotatoes, RedWineSauce |
| 4 | ChickenFillet, Salad, NoSauce |
| 5 | BeefSirloin, Salad, BoiledPotatoes, NoSauce |

(b) The contents of the results.

| # | Grapes |
|---|---|
| 1 | Zinfandel |
| 2 | Malbec |
| 3 | CabernetFranc |
| 4 | Zinfandel |
| 5 | Merlot, CabernetFranc |

(c) The grapes present in the wines.

Table C.26: Test results course 12

| Course 13 - Greek salad | | | |
|---|---|---|---|
| **#** | **Similarity** | **Wine Suggestion** | **Used in course** |
| 1 | 0.41 | Zenato Valpolicella Classico Superiore 2007 | Turkey steak with sweet potatoes, ruccola- and cheese-butter |
| 2 | 0.41 | Duca del Frassino Valpolicella Superiore 2007 | Baked coalfish |
| 3 | 0.27 | Vidigal Reserva 2005 | Chilicannelloni |
| 4 | 0.13 | Vidigal Reserva 2005 | Coq au wine |
| 5 | 0.13 | Bourgogne Couvent des Jacobins 2004 | Stuffed reindeer steak |

(a) The case results.

| # | Meal contents |
|---|---|
| 1 | TurkeySteak, Salad, SweetPotatoes, NoSauce |
| 2 | Coalfish, Tomatoes, Garlic, Onion, Pepper, NoSauce |
| 3 | VegetarianDish, Salad, Bread, TomatoSauce |
| 4 | Cock, Mushroom, BoiledPotatoes, Vegetable, Rice, Bread, GravySauce |
| 5 | Reindeer, Carrots, Onion, Vegetable, RedWineSauce |

(b) The contents of the results.

| # | Grapes |
|---|---|
| 1 | Sangiovese, Molinara, Rondinella, Corvina |
| 2 | Molinara, Rondinella, Corvina |
| 3 | Tempranillo, TintaMiuda, Malbec |
| 4 | Tempranillo, TintaMiuda |
| 5 | PinotNoir |

(c) The grapes present in the wines.

Table C.27: Test results course 13