Lung Cancer Assistant: A Hybrid Clinical Decision Support Application in Lung Cancer Treatment Selection

M. Berkan Şeşen

New College

Supervisors: Dr René Bañares-Alcántara, Professor Sir Michael Brady

A thesis submitted for the degree of

Doctor of Philosophy

Trinity Term 2013

Abstract

We describe an online clinical decision support (CDS) system, Lung Cancer Assistant (LCA), which we have developed to aid the clinicians in arriving at informed treatment decisions for lung cancer patients at multidisciplinary team (MDT) meetings. LCA integrates rule-based and probabilistic decision support within a single platform. To our knowledge, this is the first time this has been achieved in the context of CDS in cancer care.

Rule-based decision support is achieved by an original ontological guideline rule inference framework that operates on a domain-specific module of Systematized Nomenclature of Medicine-Clinical Terms (SNOMED-CT), containing clinical concepts and guideline rule knowledge elicited from the major national and international guideline publishers. It adopts a conventional argumentation-based decision model, whereby the decision options are listed along with arguments derived by matching the patient records to the guideline rule base. As an additional feature of this framework, when a new patient is entered, LCA displays the most similar patients to the one being viewed.

Probabilistic inference is provided by a Bayesian Network (BN) whose structure and parameters have been learned based on the English Lung Cancer Database (LUCADA). This allows LCA to predict the probability of patient survival and lay out how the selection of different treatment plans would affect it.

Based on a retrospective patient subset from LUCADA, we present empirical results on the treatment recommendations provided by both functionalities of LCA and discuss their strengths and weaknesses. Finally, we present preliminary work, which may allow utilising the BN to calculate survival odd ratios that could be translated into quantitative degrees of support for the guideline rule-based arguments. An online version of LCA is accessible on http://lca.eng.ox.ac.uk.

Acknowledgements

This project has been funded by the Clarendon and the New College Graduate Scholarships through the CDT in Healthcare Innovation Programme at the Biomedical Engineering Institute of the University Of Oxford. I am very grateful for the opportunity they provided for me to carry out this research.

I would like to thank my supervisors Dr René Bañares-Alcántara and Professor Sir Michael Brady for their continuous support and supervision over the course of the project. I would also like to thank Dr Timor Kadir for his supervisory role at the early stages of the project.

I also wish to extend my gratitude to our clinical collaborators: Dr Michael Peake, Dr Roz Stanley, Professor Fergus Gleeson and Dr Donald Tse for their inputs to the project.

It has to be mentioned that many people have contributed to the research presented herein through discussions and feedback, including: Professor Ann Nicholson, Dr Ernesto Jimenez-Ruiz, Dr Matthew South, Samson Tu, Dr Matthew Horridge, Professor Mark Musen, Professor Mor Peleg, Professor John Fox, Professor Ross Shachter, Louis Mayaud and Ioannis Vardakis.

Finally, I would like to thank my mother, Nuran Şeşen, for her unconditional love and support that enabled me to undertake this substantial task, and my girlfriend, Sarah Rudebeck, for being there for me.

I dedicate this work to my father, Süleyman Şeşen (1949 – 1991).

Berkan Şeşen

June 2013

Table of Contents

Chapter 1 - Introduction

The aim of this thesis is to develop a clinical decision support (CDS) prototype for lung cancer treatment selection, building specifically upon the English Lung Cancer Database (LUCADA). More generally, the thesis is an investigation and comparison of the strengths and weaknesses of the conventional rule-based and probabilistic approaches to CDS.

We begin by providing background knowledge on lung cancer, which we will refer to frequently throughout the thesis. We draw specific attention to the relatively low lung cancer survival rates in England and introduce the decision making process for treatment selection in (lung) cancer, namely multidisciplinary team (MDT) meetings. We identify the clinical need for decision support in these meetings which motivates this research.

Following this, we review related research in clinical decision support (CDS) at the MDT meetings, pointing out a number of limitations. Motivated by the clinical need and the lack of precedence, we set our primary research goal as building a single versatile CDS tool that combines several different decision support paradigms. We further specify the design goals for a successful CDS prototype which can aid the clinicians in coming to more informed treatment decisions for lung cancer patients. Finally, we present a demonstrator of our novel web-based CDS prototype, Lung Cancer Assistant (LCA), which has been developed to meet these design goals and conclude the chapter by summarising the contributions of our research and giving an outline of the thesis.

1.1 Lung Cancer

Cancer manifests itself through malfunctioning cells that proliferate in an uncontrolled way, not only invading the healthy tissues in their vicinity but also spreading to other parts of the body through the circulatory and the lymphatic system [1]. This spread of cancerous cells is termed metastasis. Clinically, there are more than 200 types of cancer since there

are more than 200 different cell types in the human body and any of these cells can grow into a cancer. For example, any of the many kinds of cells that form the lungs can cause a lung cancer. The type and the location of the cancer determines the disease's growth rate, its tendency to metastasise, its likelihood to produce chemicals that alter the way the body works and its responsiveness to certain treatment types [2].

If diagnosed sufficiently early, most cancers are treatable. The main treatment types for any cancer are surgery, chemotherapy, radiotherapy or a combination of these. Surgery involves resecting the tumour with a safe excision margin in order to ensure that no cancerous cells are left behind. Radiotherapy most commonly refers to external radiotherapy, i.e. teletherapy, which is the usage of local radiation aimed at a tumour to destroy it by damaging the cells' DNA and disrupting their cellular integrity. Radiotherapy is given fractionally so that the healthy cells exposed to the radiation near the tumour can recover, while the cancerous cells are unable to. There is also a less common type of radiotherapy which is given internally, i.e. brachytherapy, which involves inserting radioactive material inside or close to a tumour with the intention of degrading it through local radioactive exposure. Chemotherapy treats cancer through the administration of cytotoxic (cell-toxic) drugs into the patient's system. Cytotoxic drugs are formulated to attack and inhibit the growth of rapidly growing cancer cells. As a systemic treatment, chemotherapy can serve as a powerful modality to control micro-metastases across the body [3]. However, cytotoxic drugs attack other tissues in the body with high cell replication rates as well as the malignant cancer cells. This causes many side effects for the patient, and is associated with patient discomfort and co-morbidities [4].

Due to its agility, and the fact that many patients are only diagnosed at a later and incurable stage, cancer is a leading cause of death worldwide. In 2008, 7.6 million patients died of cancer, accounting for 13% of all deaths globally. The World Health Organisation (WHO) projects this figure to rise to 13.1 million deaths in 2030 [5]. Lung cancer is the most

common type of cancer globally and constitutes the leading cause of cancer mortality with 21% of all cancer-related deaths. Given that the 5-year survival remains lower than 15%, improving survival in lung cancer is a major challenge for modern oncology [6]. In particular, lung cancer survival in the UK is worse than North America and is among the lowest in Europe [4], [7].

In addition to these low survival figures, the National Lung Cancer Audit (NLCA) data reveals that there is a drastic geographical variation of 5-year survival outcomes between different primary care trusts in England, ranging from 15.4% to 43.7% [3]. While factors such as variations in patient demographics and healthcare infrastructure in different regions play a role in the inter-trust survival discrepancies, it is believed that the major cause is undesirable and unwarranted variations in practice between cancer trusts. Broadly speaking, unjustified variation in clinical practice is a pervasive finding [8] and the NLCA data reveals that lung cancer care is no exception.

Lung cancer usually reveals itself at a later stage and through common symptoms such as a continuous cough with occasional signs of blood, breathlessness and a pain when breathing or coughing. Due to late presentation of symptoms, the majority of lung cancer patients are only diagnosed at a locally advanced or metastatic stage with poor cure rates [6]. The type of treatment plan to be given for lung cancer depends on many different clinical variables concerning the patient and their disease. Due to the variation of the selection criteria applied by multidisciplinary teams (MDTs) in the UK, the rate of treatment with curative intent between different cancer networks differ significantly [3].

The most common patient-related variables that affect treatment decisions are the patient's general health, co-morbidities and lung capacity [9]. The patient's general health is usually evaluated by their performance status, which is a WHO-recommended scale that aims to quantify the patient's well-being and ability to carry on with daily activities. Compared to other chronic diseases and cancers, lung cancer causes the most disruption to quality of

life, which may persist for more than 5 years [10]–[12]. The most common co-morbidities that may contraindicate the prescription of certain treatments include (but are not limited to): chronic obstructive pulmonary disease (COPD), cardiovascular disease, renal failure, dementia and severe weight loss. Lung capacity and fitness are usually calculated by the one-second forced expiratory volume (FEV1) measurement, both as an absolute amount and as a percentage.

The most significant disease-related properties that affect treatment decisions include: the tumour's location, histological type, stage, and whether or not the tumour has metastasised. Location-wise, if the tumour is too close to the heart, trachea, mediastinum, oesophagus or any major blood vessels, surgical treatment may not be feasible. Histologically, there are two major types of lung cancer: Small Cell Lung Cancer (SCLC) and Non-Small Cell Lung Cancer (NSCLC). NSCLC cases constitute the majority of lung cancer patients, which can be further broken down into three major sub-types, namely: 1) Squamous Cell Carcinoma, 2) Adenocarcinoma and 3) Large cell carcinoma. SCLC cases are less common and the incidence is decreasing. In 2011, SCLC accounted for approximately 12% of all lung cancer diagnoses. SCLC spreads very rapidly and is caused almost entirely by smoking. There is also another type of cancer, Mesothelioma, which manifests itself in the pleura. It is usually caused by exposure to asbestos and is deemed as a rare disease that is very different to lung cancer and does not fall under the category of SCLC or NSCLC [4], [7], [13].

The tumour staging of NSCLC is done with respect to the TNM classification [14], which comprises the separate T, N and M descriptors. The T descriptor captures the size and position of the tumour, the N descriptor informs whether any regional lymph nodes are affected and the M descriptor indicates whether the disease has metastasised from the primary site of the tumour. These three descriptors are usually combined into a summary staging scale that takes a value from IA to IV, with Stage IA being the earliest and Stage

IV being the most advanced stage lung cancer. In 2009, a revised TNM staging version, i.e. version 7, was introduced world-wide. The current definitions of different T, N and M values and their corresponding Stage equivalents according to TNM 7 [15] are given in Table 1.1 and Table 1.2. Limited disease SCLC is characterised by tumours confined to one hemi-thorax; with potential local extension and ipsilateral supraclavicular lymph nodes allowed if they can be encompassed in a potentially curative radiotherapy volume [3]. All other SCLC is classified as extensive stage.

T	**The size and location of the tumour**
T1a	Tumour is contained, and is smaller than 2 cm across
T1b	Tumour is contained, and is between 2 to 3 cm across
T2a	Tumour is between 3-5 cm across, invasion into main bronchus or the visceral pleura
T2b	Tumour is between 5-7 cm across, invasion to main bronchus or the visceral pleura
T3	The tumour is greater than 7 cm across or has invaded one of: chest wall, mediastinal pleura, diaphragm, pericardium OR the tumour has made the whole lung collapse OR there is more than one tumour nodule on the same lobe of the lung.
T4	The tumour has invaded one of: mediastinum, the heart, a major blood vessel, trachea, oesophagus, the nerve controlling the voice box OR there are tumour nodules in more than one lobe.
N	**The degree of regional lymphatic node involvement**
N0	No cancer in the lymph nodes
N1	Cancer in the lymph nodes on the same side
N2	Cancer in mediastinal lymph nodes OR cancer in lymph nodes near the main bronchus
N3	Cancer in lymph nodes on the opposite lung OR at the collar bone or at the top of the lung
M	**The degree of metastasis**
M0	No metastasis
M1a	Local metastasis
M1b	Distant metastasis

Table 1.1: The definitions of different T, N and M descriptor stages

As a result of the UICC staging manual version 7, the TNM classification is now also the preferred method of SCLC staging. However, the previous binary staging classification of limited or extensive disease by the US Veterans Administration Lung Study Group is still in practical use [16] and most evidence is only available with respect to this staging.

The NICE Guideline update in 2011 states that surgery remains the preferred treatment option in NCSLC provided that the cancer is operable with acceptable levels of mortality

and morbidity risk [3]. However, due to the advance of cancer by the time of diagnosis in many patients, only 10% of patients can actually be resected. For SCLC, the most common treatment type is chemotherapy since the disease is very aggressive and has often metastasised beyond the lung at the time of diagnosis. Therefore, surgery is very infrequently performed for SCLC patients.

Stage	T	N	M
IA	T1a, T1b	N0	M0
IB	T2a	N0	M0
IIA	T2b	N0	M0
	T1a	N1	M0
	T1b	N1	M0
	T2a	N1	M0
IIB	T2b	N1	M0
	T3	N0	M0
IIIA	T1a	N2	M0
	T1b	N2	M0
	T2a	N2	M0
	T2b	N2	M0
	T3	N1	M0
	T3	N2	M0
	T4	N0	M0
	T4	N1	M0
IIIB	T4	N2	M0
	Any T	N3	M0
IV	Any T	Any N	M1

Table 1.2: The TNM version 7 descriptor definitions and the combined staging equivalents.

1.2 Lung Cancer MDT meetings

Prior to the 1990's, diagnostic assessments and treatment decisions for cancer patients in the UK were taken sequentially by clinicians who typically worked in isolation. The policy framework report for commissioning cancer services in 1995 [17] emphasised the importance of cancer care delivered by the NHS to be "of a uniformly high standard". This triggered an important paradigm shift, whereby the management of care for cancer patients would be managed by multi-disciplinary teams (MDTs) comprising clinical staff from several specialisations. The immediate benefit of these meetings is their ability to facilitate

collective thinking and expertise sharing, as opposed to the outdated sequential management by a series of specialists in isolation [18]. For this reason, MDTs are becoming the model of care for cancer patients worldwide [19]. In the UK, it is a legal requirement that the care of cancer patients is managed within MDT meetings. As a result, there are around 1500 cancer MDTs in the country, meeting weekly in different centres across the UK [20].

Similar to other cancers, lung cancer MDTs commonly consist of specialist nurses, consultant surgeons, oncologists, histopathologists, radiologists and a coordinator. One such team meets every Wednesday at 8.30 AM at the Churchill Hospital in Oxford. Through an observer contract with the Oxford University Hospitals NHS Trust, the author has been attending these MDT meetings and observing the routine yet extremely complex decision making processes of this highly specialised group of clinicians.

The lung MDT meetings usually take 1.5 to 2 hours and the team can discuss around 30 to 40 patients in a regular session. Although the meeting times are relatively short, the pre and post meeting responsibilities keep the clinical staff occupied for a substantial amount of time. Within the meeting, each patient is introduced orally by their responsible consultant or specialist nurse. While the medical history and the current treatment state of the patient is explained, the radiologist browses through the PACS[1] imaging system, retrieves the scans of the patient and shares the findings with the rest of the team. Likewise, the pathologists also present the relevant images and discuss their findings. The specialist nurses act as vital sources of "off the record" patient information, such as patient morale or dietary behaviour, which can affect the treatment decision indirectly. Once all data have been presented, the team comes to a consensus regarding the next stage of action based on the combined multi-disciplinary experience of the team. The majority of the cases

[1] Picture Archiving and Communication System

discussed comprise of recently diagnosed patients for whom the selection of the most beneficial treatment plan becomes the primary objective.

Judgements on cases are frequently formed by individual or shared experiences on similar cases in the past or relevant clinical guideline recommendations and clinical trial results. Apart from short-circuiting the often lengthy sequential decision process, and enabling the patients to benefit from the collective expertise of all members of the team, the MDT meetings also facilitate knowledge transfer between the members of clinical staff.

However, it is important to note that the volume of data that an MDT needs to process in order to come up with the best treatment decisions is not only vast but also comes from different sources, making its consolidation more difficult. Adding to the complexity of the situation, the team has to work under strict time constraints, which means not only having to evaluate a patient and coming up with the best treatment option but also accomplishing this in a matter of minutes. Taken together, these factors render the decision making environment in these meetings suboptimal and inherently prone to errors, primarily omission of relevant information.

While the performance of the team in operating under these circumstances is astonishing, ensuring consistency between decisions remains hard to achieve. Lamb et al carried out a systematic review of 37 major MDT studies from 1999 to 2009 with the goal of synthesising evidence on the factors that affect the quality of decision making in these meetings. They concluded that excessive workload and time pressure are the two detrimental factors that lower team morale, reduce attendance and rush decision making [19]. Likewise, Lanceley et al. identified that MDT meetings suffer from unstructured case discussion, time pressure and variability in the quality of decision making [21].

The fact is that variability in practice and poor decisions in the MDTs are bound to happen, not because clinicians are whimsical or untrustworthy individuals, but because they have to

make life or death decisions on phenomenally complex problems, under very difficult conditions and unfortunately with very little support [22]. As will be made clearer in the following sections, in this thesis we primarily aim to address this fundamental problem by employing different artificial intelligence techniques.

1.3. The Need for Decision Support in MDT meetings

Despite the challenges the clinicians face week after week and the evident need for Information Technology tools to streamline decision making, there continues to be only rudimentary computer support in most MDT meetings. In the lung cancer MDT meetings at the Churchill hospital there are two projectors, one reserved to display the radiological images (Centricity Enterprise software), and the other to display the cell pathology reports (Case Notes software) together with a basic electronic patient form (Interflex v5 Data Entry software) interchangeably.

While the existence of an electronic patient entry form is a positive step towards centrally storing data in a semi-computer-readable format, it is still an immense underutilisation of how computers could potentially ease the burden of the clinicians in coming to timely, safe and consistent decisions. There are many tools and methodologies in the fields of computer science, mathematics and informatics to help reinforce the diligence and expertise of the clinicians.

Clinical decision support (CDS) systems are computer tools that provide assistance in synthesising and integrating patient-specific information and presenting recommendations to clinicians at the point of care [23]. They have been implemented in many different clinical domains and at various stages in the care process, from diagnosis and treatment to monitoring and follow-up [24]. CDS systems support but do not automate the decision making process. As objective agents that can match patient data to medical knowledge, their motivation is to assist, rather than to replace the clinician [24], [25]. They can make

suggestions and flag up problems but it is the clinician who is ultimately responsible for filtering the information, reviewing the recommendations and deciding what action to take.

CDS systems are commonly implemented in clinical settings where the decision making process is error-prone due to the diversity of medical information and uncertainties associated with it. These two characteristics, along with the excessive workload and time constraints of the team members, mean that the MDT meeting exemplifies a clinical case which is ideal for CDS implementation. A CDS system, which can consolidate information from different sources and deal with uncertainty in a precise and mathematically sound way, can be employed to provide patient-specific and evidence-based recommendations in order to reduce the time pressures of the team, better structure the patient case discussions and ensure that errors of omission are minimised.

1.4 CDS systems in MDT Meetings

Given their suitability for the implementation of a computerised decision aide, MDT meetings have been the subject of previous research in clinical decision support.

1.4.1. Related Work

To date, MDTSuite [18] in colorectal cancer, MATE [26], [27] and OncoDoc2 [28], [29] in breast cancer, and MDT-QuIC [30] in urological cancer have been the major CDS tools which have been researched and applied in clinical practice. MDT-QuIC differs from the others in that it is not a computer program but a CDS tool in the form of a checklist that is implemented to act as a memory-guide to structure and guide case discussions. While it may sound trivial, Gawande showcases in his renowned article [31] how the implementation of even a simple checklist can transform the quality of care given in a hospital.

MDTSuite and MATE have been developed to help reduce the gap between clinical evidence and practice by facilitating the adoption of clinical guideline rules within the

MDT meetings. While they both operate on the principle of matching individual patient entries to a set of computerised clinical guideline rules in order to generate patient-specific arguments that support or oppose particular treatment options, they use different formalisms to computerise those rules. MATE uses a computer interpretable guideline (CIG) formalism named PROforma [32], while MDTSuite uses a more recent approach based on resource description framework (RDF) triplets and a domain specific OWL ontology for rule encoding.

Both MDTSuite and MATE follow a long-established approach in AI, which first derived from early expert systems research, in which developers aim to reproduce expert behaviour by programming the computer with deterministic rules that encapsulate domain knowledge [33]. In the context of CDS, [32] adopted such a rule-based approach, displaying the outcomes in an argument-based decision mode. This, over time, has become a conventional approach in various CDS formalisms, as we will discuss in more detail in Chapter 2.

As opposed to the rule-based representation adopted by MATE and MDTSuite, OncoDoc2 models guideline-rule knowledge in the form of a decision tree. As such, from the root of the decision tree, the clinician is guided by the software to input patient-specific information to traverse down to increasingly more specific decision nodes and finally reach a recommended course of treatment.

The performances of these tools are evaluated based on compliance rates with the encoded guideline rules. A common aspect in all the aforementioned CDS systems is that they focus on curative treatment recommendations, thereby excluding metastatic and recurrent patient cases from system evaluations. This is justifiable since metastatic and recurrent patients are commonly referred to palliative treatments for which the primary goal is no longer maximising survival. Additionally, at least in the field of lung cancer care, the guideline rule coverage for palliation is substantially limited compared to curative treatments.

1.4.2. Rule-based versus Probabilistic Approaches to Clinical Decision Support

Guideline-rule based CDS systems are undoubtedly important in facilitating guideline adherence. The argumentation-based decision model adopted by MATE and MDTSuite also has the benefit of laying out all treatment options clearly and making evidence explicit. However, a strictly guideline-rule based approach to CDS also has certain limitations. First, such systems are imprecise in quantifying the level of support (positive or negative) associated with different treatment options. As inherently qualitative formalisms, they are also incapable of carrying out statistical or probabilistic inference which can inform the clinicians on patient-specific survival likelihoods or treatment recommendations based on probabilistically maximising an outcome measure.

Second, the elicitation and maintenance of the rule-based domain representations of such systems are expensive and time-consuming. Realistically, covering the entire disease domain using only clinical guideline rules is currently close to impossible. Seroussi et al. state that the noncompliant cases in their studies exemplified the limits of clinical guideline rules in covering all clinical cases [28]. Similarly, in order to cater for cases which were not covered by official guideline rules, Austin chose to develop more than half their rule-base (134/243) based on specialised rules derived from interviews with local experts [18]. Knowledge elicitation is further complicated by the level of uncertainty stemming from the vague and implicit wording of many guideline documents. As the third and final shortcoming, such rule-based systems cannot cope well with missing data as we will investigate in Chapter 5.

A possible alternative to the deterministic approach for representing and reasoning with domain knowledge employed by such rule-based systems is probabilistic inference. Unlike rule-based systems, probabilistic models trained on existing patient data can provide quantified and more precise answers to clinically important questions such as “What is the

probability of survival for this patient?" and "How do different treatment decisions affect this probability?".

The inference mechanisms for models that can address these issues in a principled manner are based on probability calculus and -where applicable- graph theory. Hence, they are usually less explicit compared to an argumentation-based decision model that explicitly lists all treatment options along with arguments for or against them. However, there are exceptions, such as Bayesian Networks, which allow carrying out probabilistic inference in a visually more appealing and transparent way. Unfortunately, such probabilistic CDS systems are not commonly adopted since they rely on the availability of electronic patient data, which is still a rarity.

1.5. Thesis Objectives

Motivated by the increasing clinical need for CDS in MDT meetings and the limitations of the CDS systems that solely depend on guideline-rule based decision support, our primary goal in this thesis is to develop a single CDS system that can integrate different decision support paradigms. As summarised by Osheroff [34], a CDS system "*should be designed to provide the right information to the right person in the right format through the right channel at the right time*". There is a growing literature on the best practice for designing CDS systems. One such comprehensive review is by Kawamoto et al. [35] where they review the characteristics of successful deployments and conclude CDS systems are more likely to be successful when they a) provide information at the time and point of decision making; b) are presented automatically and fit into the workflow of the clinicians; and c) recommend actions rather than vague assessments.

Initially, we established the design goals for our prototype, based on a systematic review of the literature. These criteria evolved in time through our observations at the weekly lung cancer MDT meetings in the Churchill Hospital in Oxford and substantial feedback from

our clinical collaborators at different iterations of our system design efforts. The high level design goals in building our CDS prototype, Lung Cancer Assistant (LCA) are as listed below:

1) The system should serve as a decision aide that facilitates the clinicians' decision making process, rather than attempting to automate it.

2) In order to promote evidence-based medicine, the system should be able to provide decision support based on national and international clinical guideline rules in lung cancer care.

3) In addition to providing guideline-rule-based decision support, the system should be able to provide quantified and patient-specific answers to the elementary questions on the probability of survival and selection of the treatment plan that would maximise the chances of survival.

4) The system should not disrupt clinical workflow and should provide instant decision support, which would require fast inference times.

5) In order to streamline interoperability and facilitate sharing information with other software tools, the system should adhere to the international terminological standard in medicine, SNOMED-CT.

6) The system should be available online so as to facilitate rapid access and quick feedback from our clinical collaborators.

In addition to these, we also opted to develop our platform using open source knowledge standards and formalisms. In particular, the major problem of isolated and proprietary computerised guideline formalisms will be discussed in detail in Chapter 2.

1.6. Lung Cancer Assistant

We have developed the Lung Cancer Assistant CDS prototype, which addresses the clinical need for a more diverse CDS system in the MDTs and satisfies the major design goals outlined in the previous section. Throughout our research, we worked closely with our clinical collaborators, Dr Michael Peake (National Lung Cancer Audit (NLCA) clinical lead), Prof Fergus Gleeson (lead researcher for the British Society of Thoracic Imaging) and Dr Donald Tse (junior research fellow in the Churchill Hospital department of radiology). LCA has been implemented as a web-based and intelligent patient form that can not only display and summarise, but also interpret and perform different forms of inference on patient data with the aim of assisting the lung cancer experts in coming to patient-specific and evidence-based curative treatment recommendations for non-small cell and small-cell lung cancer patients.

1.6.1. LCA User Interface

The data item fields in the Lung Assistant user interface (UI) are distributed into different tabs in the order of the corresponding LUCADA database sections, which are explained in detail in [16]. Figure 1.1 shows a screenshot of the *Patient* tab of LCA.

The main components of the UI as highlighted and numbered in coloured boxes on the figure are:

1. Patient Search: is used for retrieving a patient record from the system. The user enters the id of a patient and clicks on "Search".

2. Search History: displays a list of the recently viewed patients. The “Guideline Rules” column indicates how many guideline rules apply to the respective patient record.

3. Similar Patients: lists the patients whose characteristics are similar to the currently viewed patient record. The "Similarity Level" column indicates how many guideline rules

the listed patient shares with the patient currently viewed. The user can view any of the similar patients by simply clicking on the respective patient in the list.

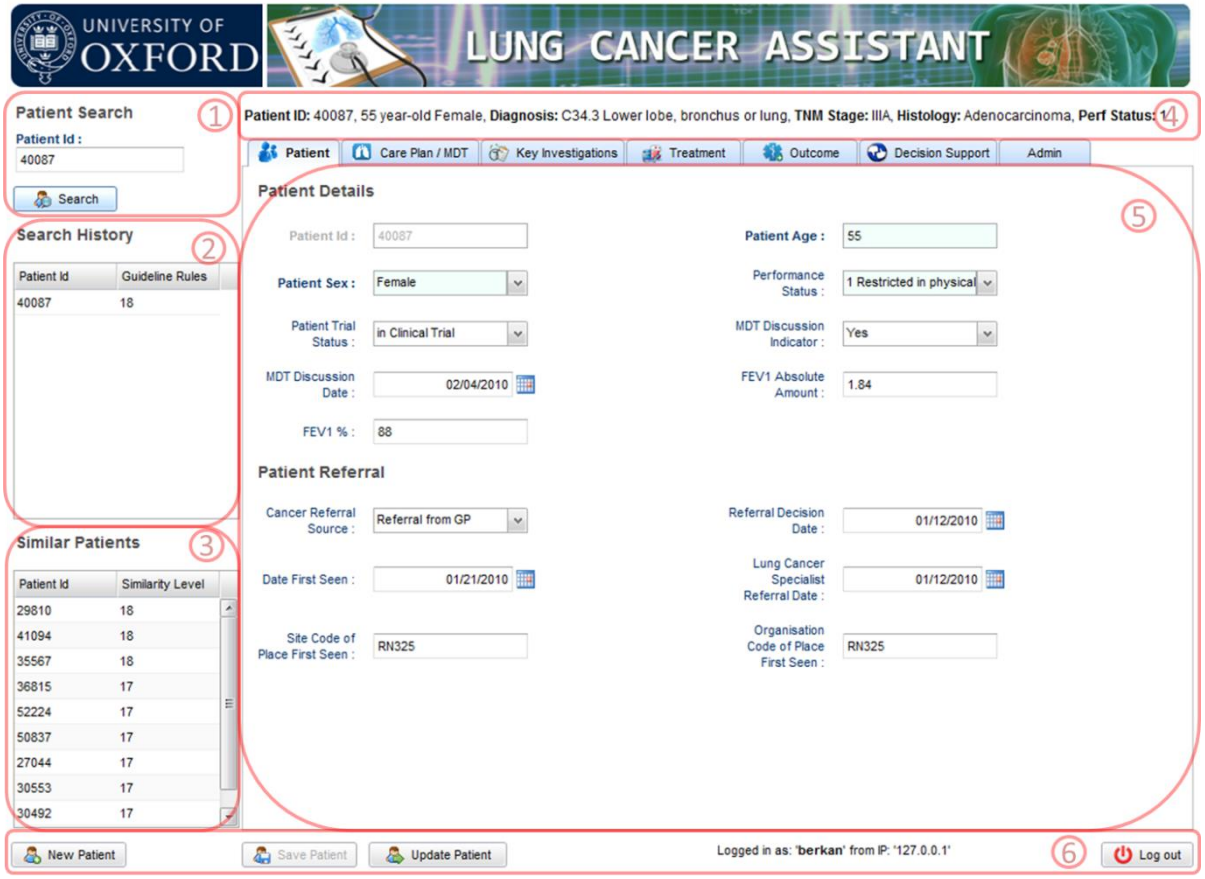

Figure 1.1: Screenshot of the LCA 'Patient' Tab. The main components of the UI are highlighted in red boxes.

4. Patient Summary: displays a brief patient summary that outlines the important characteristics of the current patient. This one-line patient summary is automatically put together from the relevant data fields of the patient record.

5. Tab Area: enables browsing through different data fields.

6. User Controls: allows updating, creating and saving patient records.

In order to facilitate and encourage compatible structured data entry, the system does not allow free text entry apart from a few numerical fields. Instead, all possible field values are presented in drop-down boxes.

1.6.2. LCA Decision Support

In addition to the six tabs explained in the previous section, LCA provides various forms of CDS under its *Decision Support* tab, a screenshot of which is provided in Figure 1.2.

LCA provides guideline-rule based decision support based on a novel ontological guideline rule inference framework that operates on a domain-specific ontological module of the international medical terminology standard SNOMED-CT. This ontological module, which is semi-automatically extracted from SNOMED-CT and further expanded to cover guideline rule knowledge, is realised in the World Wide Web Consortium (W3C) endorsed Web Ontology Language 2 (OWL-2). In order to populate our guideline rule base, we reviewed the four major clinical guideline documents on lung cancer care and, with the help of our clinical collaborators, elicited 84 treatment-selection related guideline rules.

In addition, LCA provides probabilistic decision support in order to predict the probability of patient survival and lay out how the selection of different treatment plans would affect it. This vital functionality, which cannot be realised via guideline-rule-based decision support, is provided by a causal Bayesian Network (BN) whose structure has been learned semi-automatically based on LUCADA, to which we have access through a data sharing agreement between the NHS and the University of Oxford.

Furthermore, as a convenient feature of the ontological guideline rule inference framework, when a new patient is entered, LCA can display the most similar patients to the one being viewed. This, in effect, facilitates the extraction of patient-specific knowledge derived from previous cases and acts in a way that resembles case-based reasoning operating on semantic constraint and similarity measures.

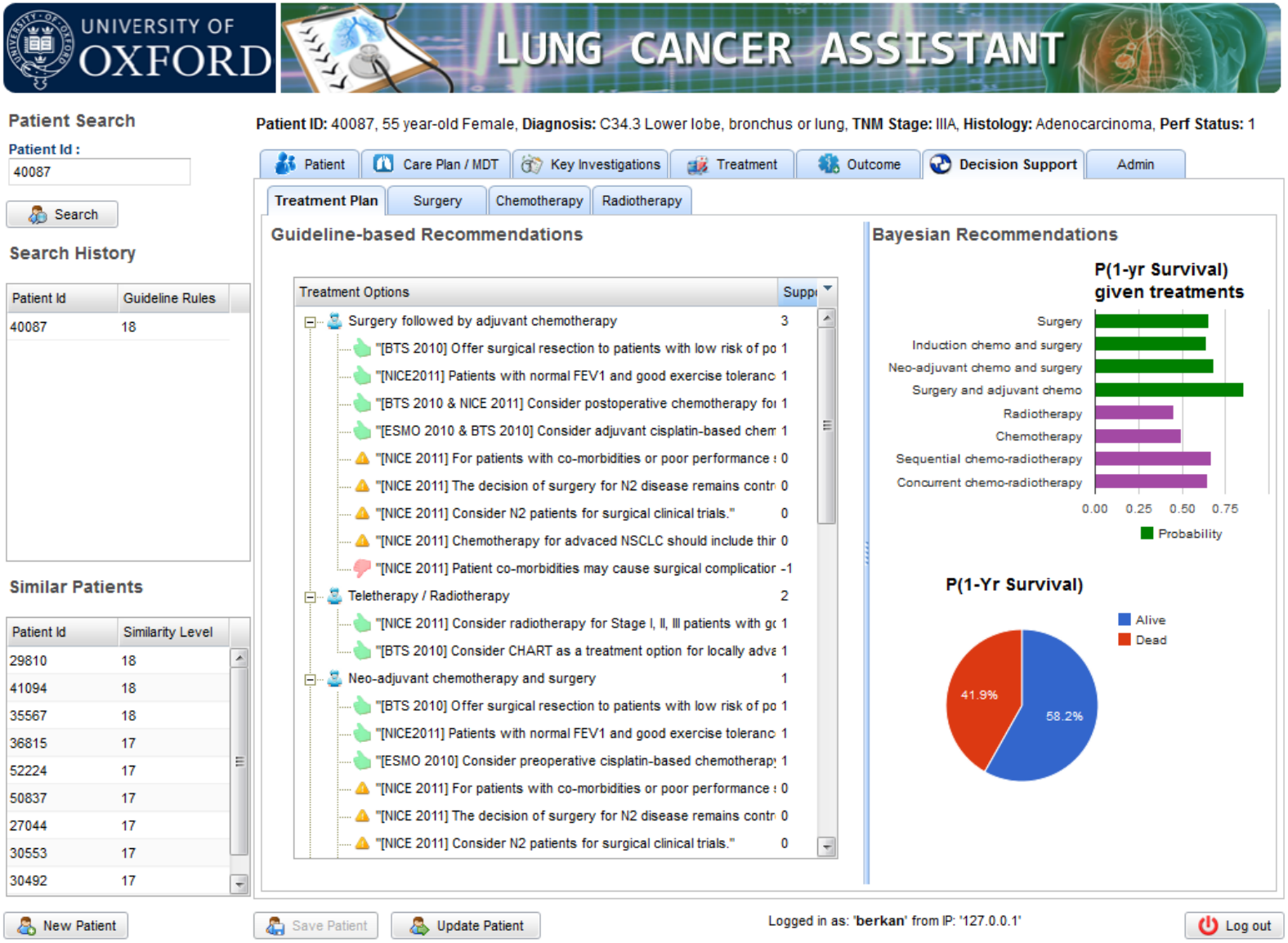

Figure 1.2: Screenshot of the LCA 'Decision Support' tab.

Integrating these different decision support techniques under a single platform, we have also carried out retrospective studies, whereby we could directly compare and contrast the strengths and weaknesses of rule-based versus probabilistic decision support on real patient data. Results of such comparisons are presented in Chapter 7.

1.7. Thesis Contributions

The contributions of the thesis are as listed below:

- We have built a guideline rule-based CDS prototype that operates on a fully ontological inference framework, making use of a highly expressive semantic language (OWL-2) to model medical knowledge based on the standardised SNOMED-CT ontology.
- In order to meet the clinical demand by the MDT members, we extended our CDS prototype to provide estimations of the probability of patient survival and how the selection of different treatment plans would affect it.
- Through retrospective experiments on a selected patient subset, we evaluated the performances of rule-based and probabilistic recommendations in predicting the recorded clinical practice, discussing the strengths and the shortcomings of both.
- We carried out an initial exploration on a methodology that would allow the automatic creation of BN queries based on the ontological guideline rule knowledge in order to appoint quantified support to rule-based arguments.

1.8. Thesis Outline

Chapter 1 has introduced the domain of lung cancer and the concept of a multidisciplinary team (MDT) meeting. It motivated the goals of our research and outlined our CDS prototype, Lung Cancer Assistant (LCA).

Chapter 2 provides background information and reviews the literature on the major knowledge representation and inference methodologies in CDS, which we made use of in order to meet our CDS prototype design criteria, set out in Chapter 1.

Chapter 3 introduces the LUCADA dataset, giving a detailed overview of all clinically relevant data fields. Furthermore, it outlines the design stages of the LUCADA ontology and its integration with SNOMED-CT.

Chapter 4 proposes a purely ontological guideline rule inference framework that is powered by the integrated SNOMED-CT LUCADA ontology presented in the previous chapter. The resulting ontological framework is then tested for scalability and inference speed to validate its applicability in a real world CDS application.

Chapter 5 describes the guideline rule base elicitation for the Lung Cancer Assistant and presents the software architecture of the first version of LCA that can only provide guideline rule based decision support. The chapter concludes with empirical results on the evaluation of the system in making meaningful guideline-based treatment recommendations.

Chapter 6 introduces an alternative approach, namely probabilistic inference, for providing decision support. It outlines the clinical need for probabilistic decision support in MDT meetings and introduces the design stages of the LUCADA Bayesian Network (BN). Furthermore, it presents experimental results on the predictive performance and Bayesian scores of various BN structures learned by different techniques.

Chapter 7 begins with assessing the performance of causal intervention queries on the BN of our choice in making plausible treatment recommendations. We compare these with the empirical results presented at the end of Chapter 5 and present preliminary work on a technique that can be used to provide quantitative degrees of support for argumentation.

Finally, Chapter 8 draws conclusions from the thesis and proposes possible avenues for further research.

1.9. Work Published from Thesis

At the time of writing, an earlier version of Chapter 4 has been published in the proceedings of the OWL Experiences and Directions workshop, which is a part of the Extended Semantic Web Conference [36]. An early version of Chapter 6 has been published in the proceedings of the American Medical Informatics Association conference [37]. Also, an extended version of the A-Box reasoning results presented in Chapter 4 has been published in the OWL Reasoner Evaluation workshop.

In addition, the BN results presented in Chapter 6 and Chapter 7 have been published in the PLOS ONE Journal. Finally, a clinical decision support paper, which describes LCA and compares the rule-based and probabilistic approaches based on the empirical results presented, has been accepted for publication in the Journal of the Royal Society Interface.

Chapter 2 - Related Work and Methodologies

In this chapter, we review the literature on various computational and mathematical tools that are used to provide clinical decision support in cancer care and research. In the first section, we introduce the most prominent Computer Interpretable Guideline (CIG) formalisms and evaluate their suitability to be used in our clinical decision support (CDS) prototype. Following our discussion of CIG formalisms, we introduce the concepts of knowledge bases and ontologies. We focus, in particular, on the Web Ontology Language 2 (OWL-2), which is the ontology language endorsed officially by the World Wide Web Consortium (W3C). Furthermore, we assess the suitability of OWL-2 as an expression language for specifying guideline rule criteria using logical expressions. Subsequently, we discuss the most comprehensive reference ontologies in medicine and cancer.

In section 3, we describe argumentation as a well-established qualitative and deterministic decision support paradigm, which is used by various CDS tools, and elaborate on its advantages and disadvantages. Following this, in section 4, we present Bayesian Networks (BNs) as an alternative - quantitative - decision support paradigm, which, in contrast to argumentation, is purely probabilistic. We also highlight the existing literature on CDS tools that make use of BNs and outline the knowledge engineering design stages in building a BN.

2.1. Guideline-based Clinical Decision Support

Clinical guidelines typically consist of a set of directions or evidence-based standards to assist clinicians on decisions about appropriate clinical procedures [38]. Adherence to clinical guidelines improves the overall quality of patient care and helps reduce practice variability and care expenses [39]. In the context of cancer care, clinical guidelines are of particular importance, since no single clinician (be it the oncologist, radiologist, surgeon or the histo-pathologist) has a comprehensive picture of the whole care pathway of a cancer

patient due to the wide ranging nature of the parameters involved in the decision making process.

However, in order to reap their benefits, clinical guidelines need to be easily accessible at the point of care by the clinicians. The sole dissemination of textual guidelines does not entail adoption and therefore has minimal impact on clinicians' behaviour [40]. The major difficulty here is that clinical guidelines are, in general, poorly structured textual documents, which are hard to interpret and therefore cumbersome to adapt in daily patient care. In addition, rapid access to these large documents at the point of care is not always easy and therefore the medical knowledge they encapsulate is not readily available. Cabana et al. list the major factors limiting the physicians adherence to clinical guidelines as " the lack of awareness of the guideline's existence, lack of agreement, lack of outcome expectancy and the inherent difficulty in changing daily practice habits" [41].

These problems can be addressed by CDS systems that computerise and automate the daily management of guidelines through computer interpretable guideline (CIG) formalisms. CIG formalisms enable encoding and interpreting guideline rule eligibility and decision criteria in order to deliver situation specific recommendations to the clinicians [42]. Many studies have shown that CDS systems that make use of CIG models can be effective in increasing clinician compliance with clinical guidelines [43]. This formalised approach has the potential of improving the acceptance, maintenance and daily application of clinical guidelines by providing guideline-based recommendations for clinicians and monitoring the decisions and actions taken. As a matter of fact, such systems have been proposed in many different medical domains including cancer, stroke and intensive care [39].

2.1.1. Computer Interpretable Guideline Formalisms

In practice, CIG formalisms link guidelines and medical concepts to electronic medical records. Although there are many different CIG formalisms in the literature, only a handful

of them have actually progressed beyond the prototype stage and made it to real life implementations. This section will give a brief overview of the commonly used CIG formalisms, namely the Arden Syntax, GLIF 3, Asbru, EON, SAGE and PROforma. These six formalisms are included on the basis of their relevance to our task-in-hand and their high impacts in the development of guideline-based decision support in medicine. However, we recognise a number of other approaches such as: PLAN [44], GUIDE [45], PRODIGY[46] and GLARE [47]; but there is insufficient room to discuss them all in more detail.

Some of the formalisms we introduce focus on the development of guidelines and their utilisation in CDS systems, while others focus more on the standardisation of guidelines and their interoperability with other clinical information systems. As we reiterate in Section 2.1.1.7, all of these CIG formalisms share common features that enable them to specify rule eligibility criteria, manage work flow and articulate medical abstractions. However, before we lay out the common features of CIG languages, it is worthwhile to give a brief overview of the ones listed above and identify the desirable features and the shortcomings of each as potential candidates for use in our CDS prototype.

2.1.1.1. Arden Syntax

Arden Syntax is probably the most well-known language for representing clinical knowledge in decision support systems [48]. It incorporates the representation of rules in terms of Medical Logic Modules (MLM) and is suitable for "sharing simple modular and independent guidelines", such as independent rules. Due to its lack of workflow management capabilities, the Arden Syntax is not suitable for modelling multistep guidelines such as treatment plans and protocols [39].

In addition, the mapping of the MLM variables to an institutional EMR requires incorporating local codes in a non-standard way [42]. Furthermore, as an aged language, it has incompatibilities with the object-oriented approaches.

Nevertheless, the Arden Syntax is one of the earliest standards accepted by the CDS community and it has been pivotal in terms providing the basis for the syntactic structures of more recent CIG formalisms such as GLIF3 and SAGE, which will be discussed in turn.

2.1.1.6. PROforma

PROforma was developed in Cancer Research UK Advanced Computation Laboratory in 1995. The language is derived from the Red Representation Language (R^2L), which is a time-oriented declarative language [49]. Like the other formalisms that will be discussed, it adopts a Task network model (TNM) approach, which consists of a set of generic tasks, such as decisions and actions that are utilised to model the control flow of a guideline.

As part of its TNM model, every guideline in PROforma is modelled as a plan that consists of a sequence of tasks that can be composed into networks representing plans or procedures carried out over time [50]. Within this setting, five types of tasks are defined to represent a guideline work flow in PROforma: 1) Keystone, 2) Plan, 3) Decision, 4) Enquiry, and 5) Action. Additionally, the details of each task and inter-relations between them are managed through logical constructs such as constraints, pre- and post- conditions and inference rules [51].

A notable aspect of PROforma is its decision model, which makes use of argumentation, whereby the decision tasks (in the TNM described above) all come with a set of decision candidates and each decision candidate has a corresponding set of arguments that are triggered by a logical condition. Each of these arguments can either support or oppose a decision candidate and are clearly visible to the clinicians to aid their decision making. We

elaborate on argumentation as a CDS approach and analyse its strengths and weaknesses in section 2.3.

PROforma comprises two different suites of authoring, testing and execution tools. Arezzo (released in 1996) and Tallis (released in 2000) both support web or standalone execution of guidelines. One of the convenient features of Tallis is that it allows web-based access to PROforma guidelines after their creation. This is a useful functionality since it not only enables the guideline developer to disseminate their guidelines with ease but also facilitates access by the end user. Unfortunately Tallis is a proprietary execution engine and is not available as open source. At the beginning of our research, we considered adopting Tallis as our CIG formalism of choice for providing guideline-rule-based decision support. However, by that time the Cancer Research UK funding for Tallis had run out and technical support was no longer available.

2.1.1.4. Asbru

Asbru has been developed by the joint efforts of Stanford University, the Vienna University of Technology and Ben-Gurion University. Similar to PROforma, the TNM of Asbru has been designed to represent time-oriented complex guidelines in the form of skeletal plans. Within the language, two types of plans are utilised: atomic and composite. Atomic plans represent single actions to be carried out, while composite plans are comprised of a collection of atomic or other composite plans [52].

Each plan's functionality is encoded by its attributes, namely by its "*preferences*", "*intentions*", "*conditions*", "*effects*" and "*plan body*". In Asbru, the temporal order of execution is managed through plan type statements such as "sequential", "parallel", "any order" or "unordered".

Prior to the beginning of our research, two execution engines had been developed for Asbru, namely Spock as part of DeGel framework [53] and the Asbru Interpreter as part of

the Protocure project [54]. However, the Protocure project was discontinued in 2006 and no technical support or application programming interface existed for the DeGel framework, which is proprietary software of the Ben Gurion University in Israel. Following our implementation of our CDS prototype, Lung Cancer Assistant, a new Asbru engine has been released by Shalom as part of his doctoral thesis and is currently being used in their E.U. funded project: MobiGuide [55].

2.1.1.2. GLIF3 (The Guideline Interchange Format)

GLIF3 is the third and final version of GLIF project, which was an initiative of the Intermed Collaboratory, intended to create an interchange format to facilitate translation of guidelines between different institutions [56]. Two common representations are employed in GLIF to facilitate this information sharing: the resource description framework (RDF) is used to store the guidelines, while the Health Level7 (HL7) standards are adopted as the generic patient model. In addition, GLIF allows hierarchies of medical concepts to be expressed and reasoned with by writing expressions that utilise these hierarchies [48]. This is a very useful functionality in terms of standardising the medical terminology used in guideline rule definitions and providing semantically richer expressions.

The initial expression language of GLIF3 was the Guideline Expression Language (GEL), which was based on the Arden Syntax logic grammar with some improvements such as the support for using complex data structures [42]. Later on, GELLO became the expression language of GLIF3. As an extensible object-oriented expression language, GELLO is built on top of GEL, supporting a superset of the functions available to GEL. In January 2005 it became an HL7 standard, recommended for all guideline publishers to be used as a common language to express their guidelines in machine readable format [57].

As part of the GLIF project, the GLIF3 Guideline Execution Engine (GLEE), was developed to execute guidelines represented in GEL [58]. From correspondences with its

developer, we were informed that GLEE is currently owned by Columbia University and actively used in a New York State HIV clinical education program [59]. Unfortunately, GLEE is proprietary software and therefore not available to public with an open source license. It therefore lacks an application programming interface (API) that would allow us to build our own web-based CDS system. More importantly, GLEE does not support the more recent expression language and the HL7 standard: GELLO [58].

At the beginning of our research, no execution engine for the HL7 standard language GELLO was available. In September 2011, a commercial company named Medical-Objects released a proprietary integrated GELLO developing environment that also allows the execution of GELLO programs. Despite being an HL7 standard since 2005, to this date no publicly available GELLO execution engine exists.

2.1.1.3. EON

EON was developed at Stanford University with the intention of describing clinical guidelines as a sequence of structured temporal steps in an object-oriented fashion. The TNM of EON is managed through a core guideline ontology which consists of various classes such as *Scenarios*, *Decisions*, *Actions* and *Activities* [39]. The "*Scenario*" class is of particular importance since it is used as the synchronisation component and enables a patient record's automatic entry into an appropriate plan or sub-plan [60]. Systems like GLIF, Prodigy and EON provide these entry points through specific modelling components such as a Scenario, as described here, or as a Patient State in GLIF. Other models, such as PROforma make use of expressions which refer to patient states in decision arguments or task preconditions for the same purpose.

Another useful attribute of EON is that unlike the rest of the CIG formalisms introduced in this section, the EON TNM, i.e. the Dharma Model, is designed in a non-monolithic way so that it can be extended to include additional classes in order to capture new guideline

behaviour [48]. The EON project was discontinued in 2002 but was succeeded by the SAGE system.

2.1.1.5. SAGE

The Standards-based Sharable Active Guideline Environment (SAGE) has been developed to formalise guidelines using a standard expression and to deploy these standardised guidelines across different institutes and Clinical Information Systems (CISs) [61]. SAGE builds upon previous work on guideline models and representation languages. The top level guideline specification in SAGE is achieved by using different process components, namely *contexts* (C), *decision nodes* (D), *action nodes* (A) and *routing nodes* (R). A SAGE deployment typically consists of a guideline execution engine server, a clinical information system (CIS) and a terminology server (for standard clinical terminology reference) [62]. The system uses the well-established clinical reference ontology SNOMED-CT for ontological references.

Among all the CIG formalisms reviewed in this section, the SAGE formalism was the most recent and promising technology, which was built upon previous work on guideline modelling formalisms such as PROforma, EON, GLIF3, Asbru and so on. The project, which was partially funded by General Electric (G.E.) Healthcare, was concluded in 2006. As reported by the corresponding author of the project, Samson Tu, the SAGE execution engine is privately owned by G.E. Healthcare and is not available for use even for research purposes. As a result, it has not been used by any clinical decision support applications and was out of our reach for the development of our CDS application.

2.1.1.7. Summary

The different formalisms discussed in the previous section have all contributed, within their own focus points, to the current state of the art in guideline-based decision support. The GLIF3 formalism stressed the importance of sharing guidelines among different

institutions and software systems but was not widely adopted since most researchers chose to use their own proprietary languages in guideline modelling. EON, GLIF3 and the SAGE formalisms have pioneered the ability to employ information and terminology standards such as SNOMED-CT and the HL7 patient model with the aim of facilitating interoperability. The SAGE project, being the most recent, synthesised some useful components from its predecessors, and was built with the priority of facilitating the integration of their applications into existing clinical information systems (CISs) and electronic medical records (EMRs).

The diversity of the different formalisms stems from the different research interests of different groups, who propose proprietary solutions to address the same problem. While the existence of diverse approaches is something to be celebrated, the fact that most of these formalisms, and most importantly their execution engines, remain proprietary unfortunately hinders open source collaboration and transfer of knowledge between different institutes. As mentioned, the emergence of GELLO as an HL7 standard in 2005 was a positive development. However, the only execution engine for this language has been developed in 2011 and belongs to an Australian company for commercial use.

We have separately discussed our reasons in not being able to adopt any of the formalisms above; the most common reason being the lack of an open source and non-proprietary execution engine, which could be integrated to our online CDS prototype and allow the inclusion of additional features to the guideline-based decision support process. Nevertheless, reviewing these different formalisms helped us acquire a better understanding of the strengths and weaknesses of each approach and identify the common components that are elementary for all CIG formalisms. Comprehensive reviews of these different systems systematically define these common components as: control-flow management, expression language, decision model, medical concept model, data abstractions and an execution engine [39], [48], [63].

Control-flow management

All CIG formalisms have similar Task-Network Models (TNMs) that model the control-flow of a guideline by decomposing it into component task networks [48]. With minor variations in implementation, these TNMs essentially use a list of common generic classes, such as plans, decisions and actions for modelling guidelines [38], [39]. For instance, guideline decisions are represented as decision tasks in PROforma, decision steps in GLIF3, conditions in Asbru and decision classes in EON.

While the building blocks remain similar, the TNMs mainly differ in the organisation and management of these generic tasks. Except for Arden Syntax, in all cases discussed above, guidelines are organised as a sequence of TNM class instances, therefore enabling the representation of multi-step guidelines and workflow management. In addition, the TNM models allow the nesting of guidelines in order to have some complex guidelines to include sub-guidelines or sub-plans. Amongst all of the above, only the TNM model of EON is non-monolithic, in that it can be extended to include additional classes if necessary [39].

In the context of a CDS tool for MDT meetings, a complex TNM model is not necessary since the emphasis of the decision support is for a single meeting and does not involve a series of sequential tasks that would necessitate temporal workflow management. In line with the findings of Austin [18], the discursive environment of an MDT meeting prioritises data collection and decision making simultaneously rather than following a clinical workflow that comprises temporal relationships or constraints between actions, queries and decisions.

In this thesis, we focus on assisting the treatment decision rather than streamlining the clinical workflow outside the meeting. Therefore, similar to the MLM model of the Arden Syntax, a simpler structure to encode the eligibility and decision criteria of modular and

independent guideline rules in a logical language would meet the requirements of our CDS prototype.

Expression Language

All CIG formalisms include an expression language to specify the decision criteria, logical expressions, and to regulate the propagation of rules. Most expression languages are proprietary but support similar logical, arithmetic and comparison operators [48]. Typically, the expression languages of all support a number of first-order logic features to encode decision and guideline eligibility criteria. Isern et al. report that the lack of a "de facto standard" language for implementing guidelines makes it harder for different CIG formalisms to share and reuse information [38].

In order to address this problem, GELLO is being promoted by HL7 to be utilised by guideline publishers as the standard machine-readable language to disseminate and exchange guideline information. However, it is not widely adopted yet, especially outside the US. At the time we were carrying out knowledge elicitation for the Lung Cancer Assistant, no lung cancer guideline rules by NICE [4], BTS [7] or ESMO [6] were available in GELLO or any other machine-readable format.

For the purposes of our research, it can be concluded that we need a first-order logic language that provides all the features listed above in a structured format similar to XML or RDF. At this stage, GELLO would be an ideal option with the advantage of also being an HL7 standard. However, the lack at the outset of our research of an execution engine to interpret GELLO forced us to look for an alternative.

Decision Model

The major decision models utilised by the CIG formalisms are switch (if/else) constructs and argumentation schema [48]. Some of the tools also support decision making by calling

external functions. Table 2.1 gives a summary of different decision models and preference representation by the discussed CIG formalisms.

The argumentation-based decision model introduced by PROforma has been implemented by other CIG approaches such as GLIF3 and SAGE [64]. It is a powerful framework as briefly mentioned in Section 2.1.1.6 and is of particular interest for the purposes of this research.

		Asbru	EON	GLIF	SAGE	PROforma
Switch (If / Else)		+	+	+	+	+
Argumentation		-	+	+	+	+
External Functions		+	-	+	+	+
Preferences	Symbolic	-	+	+	+	+
	Weighted Numeric	-	-	-	-	+
	Cost Function	+	-	-	-	+

Table 2.1: A summary of the decision models and preference representation techniques of the different CIG formalisms discussed in this section

According to the argumentation-based decision model of PROforma, every decision consists of a set of candidates, each having a set of arguments with varying levels of support for that candidate. This support, which can also be negative, can be represented as real numbers or as symbolic statements to determine the preferences among alternatives. Such an argumentation-based framework is suitable for use in a CDS application in MDT meetings and has been adopted by Patkar et al. [26] and Austin et al. [18] in their previous work for displaying the relevant arguments and potentially competing claims introduced by clinical guidelines. However, as will be discussed in Section 2.3, this purely qualitative approach also has major weaknesses, some of which can be alleviated by incorporating quantified measures.

Medical Concepts & Data Abstraction

In order to conceptualise and interpret medical data, the CIG formalisms make use of data abstractions at different levels. EON, GLIF3 and SAGE formalisms have been designed to reason with taxonomical hierarchies of medical concepts, i.e. medical ontologies [48]. This

ability to use a taxonomical nomenclature in machine-readable guidelines is definitely an advantage in order to achieve interoperability and facilitate the rapid dissemination of guidelines.

Within such a framework, for instance, the concept of "*Multiple Wedge Resection of Lung*" can be described as a "*Resection*" procedure which has focus on "*Lung*" and which has intent of removing more than one "*Lesion*" of "*Lung*". Note that this compound definition requires the definitions of different atomic concepts (i.e. Resection, Lung, and Lesion) and the relations between them, such as 'has intent' and 'has focus'. In such a taxonomical hierarchy, which can also be named an ontology, a parent concept creates an abstraction for its children concepts that inherit the attributes of its parents. Unfortunately, EON, Asbru and PROforma engines (Tallis) do not support such classification hierarchies, compound concept definitions or relationships between concepts that capture medical domain knowledge. In particular PROforma's authoring tool Tallis is only capable of using a flat data structure and cannot make use of or communicate with existing medical ontologies such as SNOMED-CT. This results in the inability to use over-arching rules and expressions, and inevitably leads to substantial amounts of repetition during guideline encoding.

Execution Engine

The execution engine of a CIG formalism is the vital component that allows the interpretation of guidelines encoded in that specific format and performs actions based on guideline information and patient data [65]. Without an execution engine, the guideline formalism becomes unusable in practice. Unfortunately, the execution engines of all formalisms discussed here have either been abandoned following the grant expiration of the corresponding project or are proprietary, belonging to a company or research institute that has exclusive rights to use the engine. At the beginning of our research, we considered adopting the SAGE or PROforma formalisms. However, as noted above, the SAGE engine

was commercially owned by General Electric and no technical support was available to improve the unstable PROforma engine to incorporate the use of ontologies or probabilistic inference.

Overall, the absence of an open source CIG formalism with a standardised expression language and publicly available execution / inference engine has motivated us to consider building our own CDS prototype with an alternative expression language and inference engine.

A recent positive development in guideline-based CDS is the 'OpenCDS' consortium that aims to connect CDS researchers across different institutions "to improve patient outcomes through the effective use of standards-based, open source CDS" [66]. Based upon a set of correspondences we had with its founder, we found out that, as of November 2012, the project is still at its earlier stages but is committed to promote open-source collaboration across multiple institutions.

2.2. Ontologies and OWL-2

In this section, we first introduce the concept of an ontology and briefly summarise the types of languages for modelling and formalising ontologies. Ontologies are pivotal for adopting a standardised nomenclature for CDS systems, and facilitating interoperability with other medical informatics software.

In particular, we focus on the semantic web ontology language OWL-2, which satisfies most of the essential requirements of a CIG language for encoding and interpretation of lung cancer guideline rules in a machine readable format. Immediately related to this, we will introduce the concept of semantic reasoners and demonstrate how they can be utilised to perform inference on the ontologies in order to interpret knowledge and perform actions, similar to a CIG execution engine.

2.2.1. Ontologies

In the context of computer and information sciences, an ontology is a formal and explicit specification of a shared conceptualisation [67]. This specification includes the taxonomy of relevant concepts and their relationships. Ontologies are particularly useful in capturing domain knowledge in a generic way and providing a commonly agreed understanding of a domain, which may be reused, shared, and standardised across applications and groups [68]. They also facilitate the development and analysis of formal models of reasoning, which is both comprehensible by humans and interpretable by computers.

In order for them to be utilised by computer applications, ontologies are formalised in languages which are closer in expressive power to logical formalisms such as predicate calculus [67]. The most notable example of such logical formalisms is Description Logics (DLs), where knowledge is 'formalised' in terms of concepts, relations individuals and axioms that are used to automatically derive taxonomies and perform consistency checks [69], [70].

Concepts are often featured programmatically as classes that represent relevant entities in the domain of interest, such as, in our case, "Lung Cancer". A class can generally be categorised as primitive or defined. Primitive classes are those that only have necessary conditions to describe them, while defined classes have necessary and sufficient conditions for an entity to be a member of that class [71]. For the purposes of logical inference, defined classes are more functional than primitive ones and, as will become apparent, they form the basis of the guideline rule inference framework, which will be covered in Chapter 4.

The relations in an ontology describe the interactions between class' properties. These relations can be in taxonomical or associative forms. Taxonomical relations organise classes in a hierarchical structure, such as "SCLC" is a "Lung Cancer" is a "Cancer".

According to this, generic classes appear at the top of the taxonomy and subsume more specific classes that descend from them. Similar to class inheritance in Object Oriented Programming, the descendant classes, i.e. subclasses, inherit the features of their ancestors, i.e. super classes.

On the other hand, associative relations relate classes across the taxonomy, and can be used to describe class attributes or spatial, causal or functional relations between classes. This is termed the concept structure of an ontology. Again, similar to the class-object model in Object Oriented Programming, each class can be instantiated with individuals that inherit their properties [72]. In addition to classes and their individuals, another key aspect of ontologies are logical axioms that are used to constrain the values for classes or their individuals [70].

Apart from DL ontologies, other ontology types also exist such as vocabularies, which are defined using natural language, and frame-based systems, which are based on frames or classes that represent collection of instances. For the purposes of our research, the former type, i.e. vocabularies, have little appeal since they lack structural representation and formally defined semantics.

Frame-based systems, on the other hand, allow a more structured approach and are therefore more desirable and intuitive knowledge formalisations. This, to a great extent, derives from the similarities of frame-based modelling and object-based modelling. These have been very popular in the knowledge representation community, especially for natural language processing applications [73]. However, the main shortcoming of frame-based models compared to DL-based ones is that they only allow the representation of primitive classes and not defined ones. Therefore, they only support explicitly asserted taxonomies and do not offer any tools that can perform inference on the ontology and efficiently compute implicit subsumption relationships between defined classes and individuals.

There exist a variety of ontological languages that belong to the three categories described above and with varying characteristics in terms of their expressiveness, computational complexity and ease of use [70]. In a nutshell, frame-based systems are less expressive than DL-based approaches. The most prominent ontology languages in the late 1990's were the frame-based Ontology Inference Layer (OIL) [74] and DARPA Agent Mark-up Language (DAML) [75], which is an agent mark-up language based on the Resource Description Framework (RDF).

RDF is a standardised data interchange model on the web, based on the idea of linking unique resource identifiers (URIs) that identify a name or a resource in the form of "subject – predicate – object" expressions. Such an expression is also known as an RDF triplet [76] and can be visualised as a directed, labelled graph with two URIs that are connected by a link. Figure 2.1 gives an example of the RDF representation of the statement that 'Lung Cancer' is a sub class of 'Cancer'.

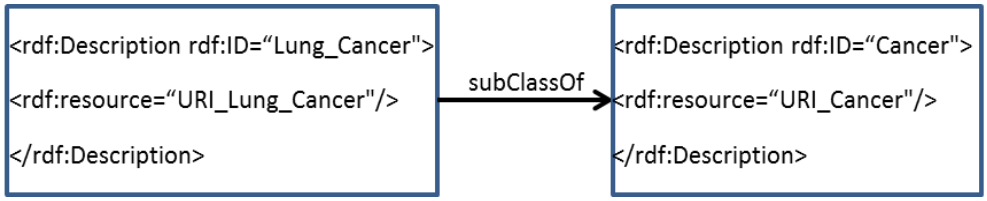

Figure 2.1: An RDF example: linking Lung Cancer and Cancer URIs

In March 2001, the Joint EU/US Committee on Agent Mark-up Languages proposed merging the features of DAML and OIL, resulting in a web ontology language named DAML+OIL [77], which was effectively a Description Logic with an RDF-based delivery syntax. Later on, the Web Ontology Language (OWL) [78] was introduced as a revision of DAML+OIL by the W3C's web ontology group.

2.2.2. OWL-2

OWL has been the most widely used ontology language since it started in 2002 [79]. As the continuation of DAML+OIL, it is based on RDF, XML and URI standards. OWL-2 is

the most recent version of OWL and is the ontology language officially endorsed by the W3C. Figure 2.2, as retrieved from the OWL-2 website [80], gives a structural overview of the OWL 2 language. As can be seen, an OWL-2 ontology can be exchanged in various different syntaxes, i.e. OWL/XML, RDF/XML and a more readable one called the Manchester Syntax [81]. The use of these structured web standards as a basis is a very desirable feature for reusing and converting expressions written in OWL-2 to other formalisms.

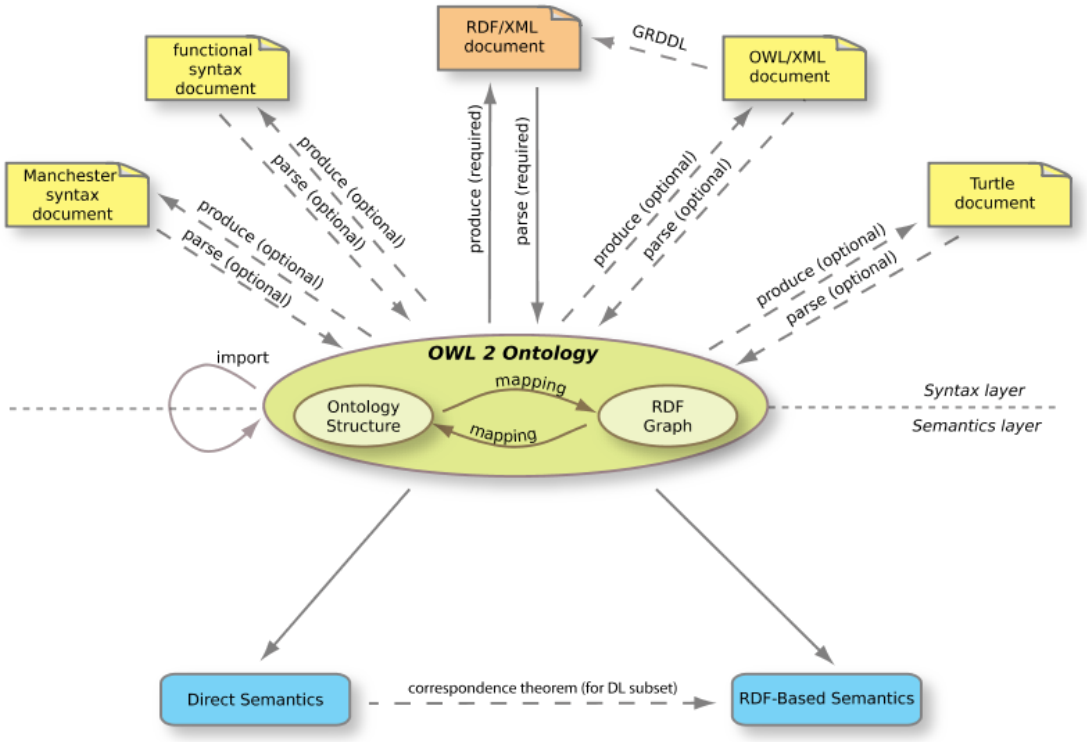

Figure 2.2: Structural overview of OWL-2 [80]

An OWL-2 ontology consists of two fundamental components: a Terminological Box (T-Box) and an Assertional Box (A-Box). The T-Box is a set of classes, properties and the respective axioms that define the constraints on a conceptual schema. The A-Box is a set of individuals belonging to the classes defined in the T-Box and is restricted by the properties and axioms of the T-Box in order to describe a particular situation [82]. In other words, T-Box is the hierarchical and conceptual view of classes and A-Box is the network view of

individuals that is based on the relationships they hold through object and datatype properties.

Object properties are relationships between two individuals, such as ‘hasClinicalFinding’, ‘hasHistology’ or ‘hasLaterality’. On the other hand, data properties link an individual to an XML Schema Datatype value or a Resource Description Framework (RDF) literal [71], [76]. In other words, they describe relationships between an individual and data values, such as, ‘Hospital ID’ or ‘Age’.

In Figure 2.3, an example of this T-Box/A-Box ontological structure is given in the context of a ‘Patient’ individual, who has ‘Lung Cancer’, with a ‘Non-small cell carcinoma’ type tumour. It is important to note that the object and datatype properties of individuals in the A-Box are directly inherited from their parent classes, and each parent class inherits these from their ancestors within the taxonomy. For instance, the ‘Hospital ID’ datatype property is directly inherited from Berkan Sesen’s parent class: ‘Patient’, whereas the ‘Age’ datatype property is indirectly inherited from ‘Person’ as a result of the “Sub class” assertion between ‘Patient’ and ‘Person’. Overall, an OWL-2 ontology is the sum of all such assertions about a domain knowledge and facts in the form of logical axioms. Electronically, it looks very much like a well-structured XML document that contains a list of logical axioms of different types.

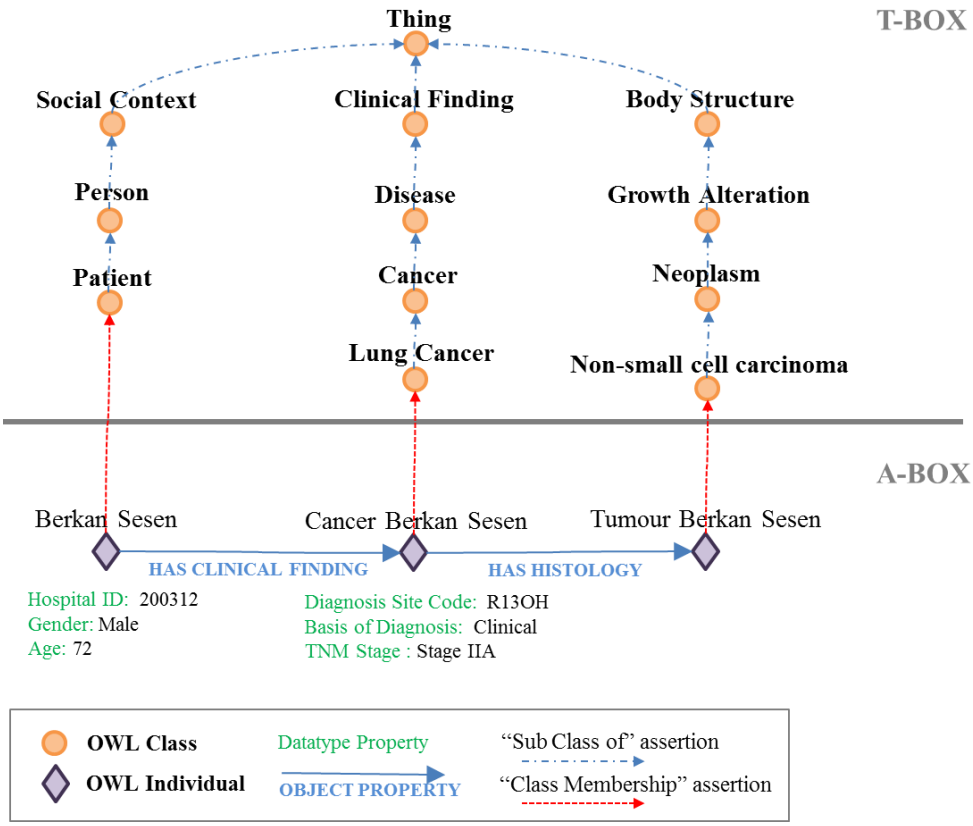

Figure 2.3: A T-Box/A-Box example within OWL-2, which models a particular 'Patient' individual having 'Lung Cancer' with a tumour of 'Non-small cell carcinoma' histology type.

However, these logical axioms need not only be in the form of class, individual and property assertions, which relate the classes and individuals to each other. An OWL-2 ontology also includes a set of axioms which restrict the sets of individuals and the types of relationships permitted between them. In OWL-2, there are four main restriction types, namely: Existential (Some) Restrictions, Universal (Only) Restrictions, 'has Value' Restrictions and Cardinality (min, max, exactly) Restrictions. These axioms provide semantics by allowing systems to infer additional information based on the asserted data. A detailed and applied explanation of these restriction types are given by Horridge et al in [71].

These restriction and assertion axioms can be used in the context of "necessary" and "necessary and sufficient" logical conditions to express class equivalence axioms. As mentioned previously, classes that have 'necessary and sufficient' conditions are termed as defined classes. Figure 2.4 shows an example of a defined class: 'Old Patient' as modelled

in the graphical user interface of the open-source knowledge representation tool Protégé 4.1 [83]. The necessary and sufficient logical condition: "**Patient and AgeDiagnosis some integer [>70]**" is placed as a class equivalence axiom in the figure. It is written in the more easily readable OWL Manchester syntax and entails that if an OWL individual is a member of the 'Patient' class and has its 'AgeDiagnosis' data property filled in with an integer greater than 70, then these are sufficient to deduce that it must be a member of the 'OldPatient' class.

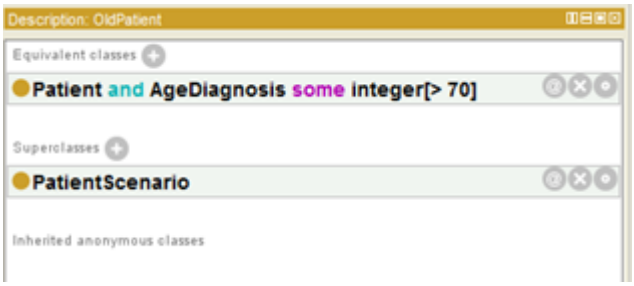

Figure 2.4: The 'OldPatient' defined OWL class with a 'necessary and sufficient' condition asserting its class equivalence. This entails that if something is a Patient and has a diagnosis age greater than 70, these conditions are sufficient to deduce that the OWL individual must be a member of the 'OldPatient' class.

As the reader may have noticed, we expressed this particular axiom by making use of a class assertion, a datatype restriction (**some integer [>70]**) and some logical constructors and arithmetic operators such as 'Union of (**and**)' and 'Greater than (>)'. It is worthwhile to mention that the datatype restriction in this example, which constrains the range of integer values allowed for the 'AgeDiagnosis' data property is a feature of OWL-2 and was not available in OWL. As a matter of fact, OWL-2 comes with many additional features to the previous version of the language, such as the inclusion of richer data types, extended datatype capabilities, data range combinations, qualified cardinality restrictions, property chains and simple meta-modelling capabilities [84]. Altogether, these make OWL-2 a more complete and expressive logical language, which can indeed be used, in a similar way to a CIG language, for encoding guideline rule criteria.

2.2.3. OWL-2 as a logical expression language

At the end of our literature review on CIG formalisms, we concluded that a CIG language and its execution engine should be able to 1) provide a standardised and structured logical language to encode clinical guidelines and form a machine readable guideline repository, 2) enable the connection of this repository with electronic health records, 3) allow the use of standardised reference ontologies in guideline encodings and 4) come with an inference engine that can interpret the logical expressions and criteria encoded in that language [38]. While various formalisms that satisfy some of these criteria exist, the absence of a well-maintained and publicly available CIG formalism with adequate technical support and open source access motivated us to look for an alternative expression language and inference engine for encoding lung cancer care guideline criteria.

As a W3C endorsed, standardised and well-established ontological language, OWL-2 satisfies all requirements listed above and is therefore a suitable candidate. Semantically, it utilises the SROIQ description logic, which is a fragment of first order logic with useful computational properties [80]. Furthermore, it defines several language subsets, i.e. profiles, which are aimed to better meet certain performance requirements of specific user cases. As we demonstrated briefly with the 'OldPatient' example in the previous section, OWL-2 provides logical and arithmetic operators like all CIG expression languages, which we can utilise for the purposes of encoding clinical guideline eligibility and decision criteria. Table 2.2 lists the most common OWL-2 constructors.

OWL and OWL-2 are also commonly utilised for formalising and maintaining standardised reference clinical ontologies, such as UMLS and SNOMED-CT. As a result, an inherent advantage of using OWL-2 for encoding guideline rule knowledge is the convenience it brings in seamlessly integrating the guideline knowledge with the established domain knowledge in such reference ontologies.

Constructor	DL Syntax	Manchester Syntax Example
Intersection of	$C_1 \cap \ldots \cap C_n$	Patient **and** Female
Union of	$C_1 \cup \ldots \cup C_n$	Male **or** Female
Complement of	$\neg C$	**not** Female
all Values From	$\forall P.C$	hasClinicalFinding **only** Cancer
some Values From	$\exists r.C$	hasHistology **some** Carcinoma
has Value	$\exists r.\{x\}$	HospitalID **value** "200312"
min Cardinality	$(\leq n)$	hasChild **min** 3
max Cardinality	$(\geq n)$	hasChild **max** 3
exactly	$(= n)$	hasChild **exactly** 3

Table 2.2: The most commonly used OWL-2 language constructors, where 'C' is an OWL class, n is an integer and x is an XSD facet.

We will introduce the most prominent clinical reference ontologies in Section 2.2. For the time being, it should suffice to say that integration of our guideline knowledge with such ontologies would help with the dissemination of our knowledge to other institutions and greatly facilitate interoperability with clinical software tools that adopt the same reference standards.

In addition, there are well-maintained, free and open source graphical user interfaces for authoring OWL ontologies, the most common being the Protégé [83] ontology editing and knowledge representation platform, which has been developed and is being maintained by the Stanford Biomedical Informatics Research Group. Protégé is written in Java and provides a modern and intuitive graphical user interface for designing and rapid prototyping of ontologies in a variety of formats, including OWL, RDF schema and XML schema. As an intuitive and mature user interface, Protégé has been used to represent domain knowledge in the GLIF and SAGE projects by Stanford University. It is a very convenient tool for manually designing OWL-2 ontologies, which can then be used in software tools that make use of these ontologies.

Furthermore, OWL and OWL-2 come with an open-source application programming interface (API) in Java in order to programmatically access and modify OWL ontologies. This is called the OWL API [85] and is primarily maintained by the University of Manchester. The OWL API provides the necessary open-source tools which can be used to

build OWL / OWL-2-based ontological applications without having to be dependent on a proprietary formalism or software. The versatile nature of the API allows developing software that can easily connect with patient records in a database. More importantly, both the OWL API and Protégé have global user communities and support forums, through which independent software programmers can interact and assist each other's efforts. Especially in the early phases of the project, these forums have been very helpful in terms of getting answers to technical questions and also building up contacts in the OWL community.

Finally, as an ontology language, inference for OWL-2 is performed by semantic reasoners that infer logical consequences from the set of asserted axioms and facts in the ontology. Due to the popularity of both OWL and OWL-2 languages, many reasoners are freely available that can interpret the logical axioms in these languages. In the next section, we will discuss such reasoners and list the most commonly used ones.

2.2.4. OWL-2 Semantic Reasoners

Probably the most powerful characteristic of OWL and OWL-2 ontologies is that the set of axioms they hold can be processed by DL reasoners to provide inference. One of the main inference services provided by a reasoner is consistency checking, which ensures that an OWL-2 ontology does not contain any contradictory axioms.

Another important inference service is "classification", whereby the reasoner computes the subclass relations between all classes in an ontology and returns a complete 'inferred' class hierarchy [86]. At this point, it is worthwhile to note that ontological classification should not be confused with classification in the context of machine learning, which is concerned with identifying to which of a set of categories a statistical observation belongs.

Both (ontological) classification and consistency checking are helpful tools for the design and maintenance of ontologies since they ensure the logical integrity of the ontology at

hand, pointing out systematic modelling errors, contradicting logical axioms and missing relationships [87].

Related to classification, another commonly used inference service is called "realisation". It is used to "compute the direct types for all individuals" [86], or in simpler terms to return the inferred class memberships of the individuals in the ontology. Since these direct class memberships are defined with respect to the class hierarchy, realisation can only be performed after classification.

As an illustrative example, let us assume that in our ontology we have a 'Patient' class and a defined 'Old Patient' class with the equivalent class axiom as shown in Figure 2.5. Consequently, we create a 'Patient' individual named 'Berkan Sesen' and fill in its 'AgeDiagnosis' datatype property as 72. At this stage, when we perform classification and realisation on the ontology, the reasoner infers that 'Berkan Sesen' has the "necessary and sufficient" conditions and realises it as an individual of 'Old Patient'. In Chapter 4, we will further investigate how 'realisation' is utilised in more complicated cases for guideline rule eligibility criteria encoding.

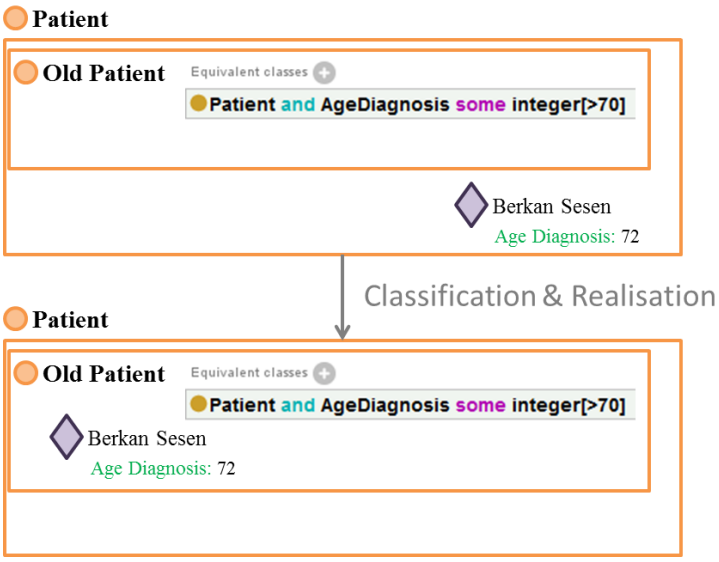

Figure 2.5: The semantic reasoner automatically infers that the newly added Patient individual is a member of the Old Patient class.

In order to take advantage of these inference services, OWL and OWL-2 users can make use of numerous different semantic reasoners. A complete list can be accessed from the W3C's website [88]. The most common OWL-2 reasoners are FaCT++ [82], HermiT [89], [90] and Pellet [86], [91]. These are all general purpose SROIQ logic reasoners that can deal with the majority of the use cases in terms of expressivity and the range of axioms they support. In addition to these, a commercial reasoner named RacerPro [92] also exists.

There are also profile-specific reasoners (OWL-2 QL, EL,and RL) that are tailor-made to reason with ontologies which can only process predefined subsets of OWL-2. A few of the better known ones for the OWL-2 EL profile can be listed as CEL [93] and its Java-implemented version JCel [94], ELK [95] and Snorocket [96].

All the reasoners listed here are capable of carrying out the aforementioned elementary inference services, albeit with varying range of axiom support and performance. The main method to perform inference on an OWL-2 ontology is to load the ontology and then process it through one of these reasoners, which can be accessed either through semantic reasoner plug-ins within Protégé user interface or programmatically by their API's that are compatible with the OWL API in Java.

The availability of such inference tools is indeed a great luxury for application developers and has contributed greatly to the increasingly widespread use of OWL and OWL-2, not only in Semantic Web but also as a very popular language in the fields of medicine, biology, geography, engineering, defence, astronomy, agriculture and so on [97]. In the next section, we will briefly discuss the domain specific reference ontologies in medicine and cancer.

2.2.5. Reference Medical Ontologies

Like many intricate fields of medicine, cancer treatment and research require managing an ever-growing flood of knowledge from different disciplines, which cannot be handled with

traditional approaches whereby individual clinical and research efforts proceed in relative isolation with their own terminologies and information systems [98]. As a result, researchers increasingly focus their efforts on the standardisation and consolidation of the widening net of terminology and relations used in the multidisciplinary domain of cancer. As discussed in the previous section, ontologies are perfectly suitable for both representing and reasoning with this rapidly expanding knowledge.

Medical ontologies in general can be put into two main categories. The first category holds narrow, subject-specific ontologies which model a domain of interest in detail. These ontologies are usually tailor-made knowledge-bases designed to be used in a proprietary application. The second category contains high level reference ontologies, which aim to describe general medical concepts in a comprehensive but non-detailed manner. These reference ontologies can be used as convenient starting points to branch off lower down to create the domain specific ontologies of the first category. As a matter of fact, we will showcase a hands-on implementation of this vital ontological design heuristic in Chapter 3.

The most prominent reference medical ontologies, which are actively maintained and are suitable for use in a cancer care CDS prototype, are the UMLS Metathesaurus [99], Foundational Model of Anatomy [100], National Cancer Institute Thesaurus [101] and SNOMED-CT [102], [103]. A comprehensive list of other commonly used medical ontologies can be accessed online at the BioPortal [104] website, which is maintained by the National Centre for Biomedical Ontology.

The UMLS Metathesaurus is a large, multi-purpose vocabulary database, which is built from the electronic versions of numerous other medical classifications, code sets and ontologies. The NCI thesaurus and SNOMED-CT are among the source vocabularies of the UMLS Metathesaurus [105] . More information on all UMLS knowledge sources can be found in [106].

The Foundational Model of Anatomy [100] is developed as a reference ontology of anatomy that can be used by any computer application making use of anatomical knowledge. It is very detailed in terms of the properties and relationships that exist in the ontology. However, as the name suggests the domain is limited to anatomical concepts alone, and is therefore very niche.

The NCI Thesaurus is a public domain 'description logic-based' terminology produced by the National Cancer Institute with "the goal of providing a controlled vocabulary that can be used by specialists in various sub-domains of oncology" [107]. It attempts to combine features of both high-level and domain-specific ontology efforts by providing a generic logic-based reference terminology.

SNOMED-CT is a clinical terminology that provides clinical content and expressivity for electronic healthcare services. It is owned, maintained and distributed (freely) by the International Health Terminology Standards Development Organisation (IHTSDO), a non-profit global association in Denmark. It is the most comprehensive medical ontology, with over 344,000 active concepts that cover most areas of clinical practice such as diseases, findings, procedures, microorganisms, medications, etc. [108]. It is comprised of concepts, terms and relationships with the objective of precisely representing clinical information. The concepts in the ontology have unique meanings and formal logic-based definitions, organised into taxonomies.

NHS Connecting for Health, which will become obsolete in July 2013, is the UK license granter for SNOMED CT. In August 2011, SNOMED-CT was approved by the Information Standards Board as the full fundamental standard for all medical information applications within the NHS. On their website [108], NHS Connecting for Health declares that "*SNOMED CT is the chosen terminology of the NHS in England. All clinical computer systems within England will operate using SNOMED-CT as the clinical terming/coding standard and will replace or incorporate other code systems currently in clinical*

medicine". There are also e-learning courses intended to give clinical staff a basic understanding of what SNOMED-CT is and how it is structured. For the purposes of adopting a standardised ontology to be used in our CDS prototype, SNOMED-CT is an obvious choice due to its wide coverage of medical concepts and its official adoption by the NHS.

2.3. Argumentation

It is common for incomplete or inconsistent knowledge to arise in many fields of science. Cancer care and research are prime examples of such fields, where the level of uncertainty is further amplified by the rapid discovery of new knowledge. Argumentation is an emerging research field that focuses on building formal methods for handling uncertainty and conflicting information. Argumentation provides an intuitive solution to dealing with uncertainty by constructing and evaluating arguments and counter-arguments for conflicting information sources.

Caminada et al. [109] define argumentation as a four stage process which consists of 1- Argument Construction; 2- Conflict Detection; 3- Determining the Acceptability of Arguments and 4- Deciding on Justified Conclusions. From the perspective of a CDS application, Step 4 is not only superfluous but also undesirable since it automates decision making by resolving the conflicts between different arguments and coming to a conclusion without any human interaction. While fully automated decision-making may be suitable in some application domains, (e.g. process control, credit score check, insurance liability calculation, etc.), it is not compatible with CDS where the aim is to assist the clinician in their decision-making, rather than taking full control.

In the context of CDS, the argumentation-based decision model proposed by the PROforma CIG formalism [32] is one of the earliest and most commonly adopted frameworks. According to this approach, clinical guideline rules, which are composed in a

machine readable format, are evaluated against a patient record in order to construct arguments that support or oppose existing treatment options [110]. These patient-specific arguments, which are in favour of or against a given treatment option, are then aggregated by the CDS application to compare different treatment options and recommend to the user the treatment that has the most supportive arguments. This is an intuitive way of formalising a pro / con list for different treatments by consolidating information from diverse sources and as a result has been successfully adopted in various CDS prototypes for use in MDT's [18], [26].

In 2009, Gorogiannis et al. proposed a defeasible argumentation framework for reasoning about clinical trial results by constructing arguments for and against the use of treatments with specific patient classes, based on specific clinical outcome indicators [111]. Their purely theoretical framework investigated the notions of attacks and defeasibility between different arguments that have contradictory claims, with the overall aim of drawing inferences from sets of clinical trial results that are incomplete or inconsistent. Compared to the framework of PROforma, which makes use of first order logic, their approach used a more simple and less expressive logic that was developed by the authors to capture and represent clinical study results as rules. Despite being a significant contribution to the field, as the authors conclude, the feasibility of this proposal is yet to be validated in a practical implementation.

From our perspective, a major shortcoming of these argumentation-based frameworks is that an argument's support can only be represented qualitatively. Within PROforma's decision model, arguments can either be given symbolic weights such as 'for', 'against', 'confirming', 'excluding', or they can be given arbitrary quantitative weights, e.g. '+100', '-10,000' appointed by the system developer. In the defeasible framework proposed by Gorogiannis et al, a similar approach is used, whereby the system developer indicates qualitative comparisons between treatment options as: Treatment 1> Treatment 2 for

specific patient groups. In reality, these arbitrary support indicators substitute for formal and statistically sound methods in the absence of directly relevant patient data. We believe that appointing a decisive weight or comparative utility for a treatment plan should be backed up by rigorous statistical evidence in addition to the qualitative judgement of an expert.

The proponents of argumentation either choose not to dwell explicitly on this issue or advocate that such qualitative comparisons are suitable for use in clinical practice since they capture "*a common-sense way of reasoning with conflicting information*" [111] and "*clinicians employ clinical knowledge in a very different, non-probabilistic way*" [112].

Although a qualitative judgement of an argument is valuable, a quantitative and more precise measurement of the degree of credibility or the support of an argument may be more informative and help to decide whether a hypothesis proposed by the arguments can be believed or not. For this purpose, a probability structure imposed on the arguments may help quantify the reliability of these arguments and introduce degrees of support for hypotheses [113]. In the next section, we will look into Bayesian Networks, which can provide such a quantifiable probability structure.

2.4. Bayesian Networks

A Bayesian network (BN) is "*a graphical model of a joint probability distribution over a set of random variables*" [114]. It enables reasoning with an uncertain domain [115]. The generality of the BN formalism makes it useful across a wide variety of circumstances [116].

Over the last two decades, BNs have become a popular representation for encoding uncertain domain knowledge, especially in healthcare and biomedicine [116], [117]. They are suited for clinical applications due to their ability to model causal interventions and to reason both diagnostically and predictively [115]. Part of their popularity stems from their

visual appeal, which renders them easy to analyse and modify by domain experts. Murphy defines a BN as "a marriage between probability theory and graph theory" [118]. A BN consists of two components [119]:

1- A directed acyclic graph (**DAG**): A DAG is a directed graph without cycles and is made up of a set of nodes (N) and a set of arcs (A) [116]. We can denote DAG= <N, A>, where N represent random variables that can take on different states and A model the causality relationships between different nodes. Informally, an arc from node n_x to node n_y means that n_x is a parent node (Pa) of n_y and that "n_x influences n_y" [115], [116]. The absence of an arc between two nodes means that the corresponding nodes do not influence each other directly and are therefore probabilistically independent. Effectively, a node n_x is "dependent on only its parents and its children but is conditionally independent of its-nondescendants, given its parents" [114]–[116]. This is called the 'Markov Condition'. The DAG of a BN provides an intuitively appealing interface by which humans can model interacting sets of variables within a data structure suitable for designing efficient general purpose algorithms [118]. Figure 2.6 gives an example BN DAG.

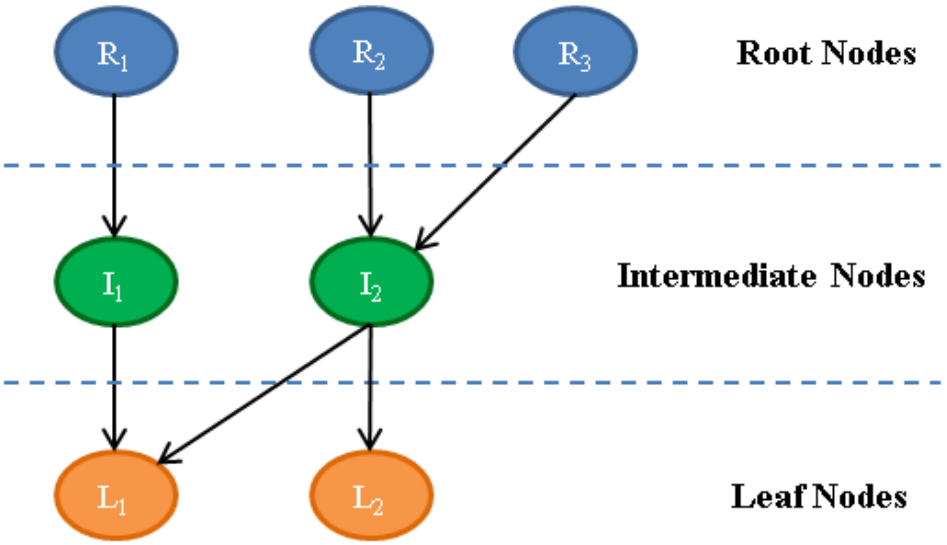

Figure 2.6: An example Bayesian network with root, intermediate and leaf nodes.

2- A joint probability distribution: Associated with every DAG is a joint probability distribution which is specified by a set of parameters ϴ. For each node n_x in a DAG, a

conditional probability distribution (CPD) is defined as $P(n_x|Pa_{nx})$, entailed by the Markov Condition [115]. Each of these distributions describes the joint effect of a specific combination of node states for the parents Pa_{nx} on the probability distribution over the states of n_x. As a result, the overall joint distribution over a probability space can be calculated as a topological factorisation of the constituent conditional probability distributions. In the set of equations (2.1), the top equation gives the full factorisation and the bottom equation gives the factorisation with the Markov assumption.

$$\mathrm{P}\left(n_1, n_2, \ldots n_x\right) = \prod_{i=1}^{n} P\left(n_i \mid n_1, ..., n_{i-1}\right) \quad \textbf{(2.1)}$$

$$\mathrm{P}\left(n_1, n_2, \ldots n_x\right) = \prod_{i=1}^{n} P\left(n_i \mid Pa_{ni}\right)$$

Using the Markov assumption, the joint probability for the BN given in Figure 2.6 can be more compactly represented as:

$$P\left(L_1, \ldots, R_3\right) = P(L_2 \mid I_2) \bullet P(L_1 \mid I_1, I_2) \bullet P(I_1 \mid R_1) \bullet P\left(I_2 \mid R_2, R_3\right) \bullet P\left(R_1\right) \bullet P\left(R_2\right) \bullet P\left(R_3\right) \quad \textbf{(2.2)}$$

It may be observed how the Markov condition helps reduce the amount of probabilistic information and the size of Θ that has to be elicited to form the overall joint probability distribution of the BN in Figure 2.6. Provided the number of parents for each node is finite, the number of parameters required grows only linearly with the size of the network, whereas, in general, the joint distribution without assuming the Markov condition grows exponentially [120].

2.4.1. Bayesian Networks in Clinical Decision Support

As generative probabilistic inference tools, BNs are particularly suitable for CDS applications since:

1) their graphical nature provides a visually more appealing and transparent inference mechanism compared to ‘black-box’ modelling techniques [121].

2) they can formally incorporate diverse types of evidence, including domain knowledge and prior beliefs, during structure learning and parameterisation [122].

3) they reduce the burden of parameterisation due to their compact representation of a joint probability space as given in (2.1).

4) they allow explicitly modelling causal interventions [115].

5) they can be used to perform probabilistic inference both diagnostically (effect to cause) and predictively (cause to effect) for any given node in the network.

6) they are better in making predictions with incomplete data [123], [124] due to the causal Markov assumption.

As a result of these desirable features, BNs have been used in clinical decision support for a variety of purposes, such as disease diagnosis, optimal treatment selection, construction of epidemiological disease models, interpretation of microarray gene expression data and so on [114]. Among others, disease diagnosis seems to be by far the most common BN application area in CDS. One of the earliest medical implementations of BNs in medicine is the ALARM network [125] that was used to monitor patients in the intensive care unit. More recent applications include a probabilistic model for diagnosing liver disorders [126], the oesophagus network [127] for aiding in the diagnosis of oesophageal cancer, a CDS tool [128] for modelling the uncertainties associated with mammography diagnosis, Take Heart II [129] for cardiovascular risk assessment and most recently the Promedas [130] commercial platform that allows incorporating BNs in creating bespoke diagnostic systems. A more complete list of BN applications in medicine can be accessed from Table 5.2 in [115].

2.4.2. Bayesian Network Design Stages

BNs can be designed either by eliciting expert knowledge or by automated search through specialised algorithms. The design stages described in this section are similar to those

proposed within the Knowledge Engineering with Bayesian Networks (KEBN) lifecycle model by Boneh [131]. As a knowledge engineering effort, design and deployment of BNs is an incremental and iterative process that contains various distinct steps, namely variable selection, structure learning, parameter learning and inference.

2.4.2.1. Variable Selection

There are two competing goals when designing a BN. On one hand, we work towards maximising the fidelity of the BN by making sure that no relevant variables or variable states are left out in our model. On the other hand, a higher number of variables results in a higher number of joint probability parameters, adding complexity and computational overhead to the structure learning, parameterisation and belief updating tasks. The curse of dimensionality becomes a problem as the probability space expands with the number of variables and variable states. Especially in cases when a particular combination of parent values of a node is uncommon, the data available to estimate the conditional probabilities become very sparse. Therefore we have to make sure that all superfluous variables are omitted in order to keep the joint probability space as compact as possible.

In other words, a trade-off between the BN's faithfulness and the cost of additional modelling is necessary when determining which variables to include in the BN. As in all machine learning tasks, we start by designating some outcome variables that we wish to query the BN about and some predictor variables that are observed and input as evidence. In clinical applications, expert elicitation is usually crucial to determine which evidence variables are the most relevant to the prediction problem. Although it would be naïve to assume that clinicians have full and unbiased knowledge of how all different domain variables affect each other, they are the ultimate decision makers and can at least provide reliable information on which variables are more informative for the prediction task and should therefore be included within the BN model.

2.4.2.2. Structure Learning

The aim of the structure learning is to find the DAG that best fits the dataset and represents the domain. Learning the structure of a BN is a specific manifestation of the more general problem of selecting a probabilistic model that explains a dataset. Robinson [132] has shown that the number of possible DAG structures for a BN with n nodes is given by:

$$r(n) = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} r(n-i) = n^{2^{O(n)}} \quad \textbf{(2.3)}$$

As a result of the exponential size of the search space, this equation gives r(1)=1, r(2)=3, r(5) = 29,281 and r(10) ≈ 4.2×10^{18} [133]. Chickering et al. have proven that learning the structure of a BN is an NP-hard problem [134].

In general, structure learning algorithms can be categorised into 1) Constraint-based approaches that use conditional independencies and 2) Score-based search approaches [116]. Most score-based structure learning algorithms involve implicit parameter learning as part of their processes. As a direct consequence of the 'dual' nature of a BN, these algorithms proceed to first learn a DAG and then parameterise it. These references cover the topic of learning structures in substantial detail: [117], [135], [136].

The constrained based methods make use of statistical independence tests such as Chi Square or the two-way likelihood ratio (G2) test to find conditional independencies in the data. The two most commonly used constraint-based methods are the Inferred Causation (IC) [137] and PC [138] algorithms, which are applicable for BNs with discrete variables. Subsequently, improved variations of both algorithms have also been proposed.

On the other hand, score-based methods are used to search for the causal model that maximises a metric score in the causal model space, which –as given in (2.3) - expands exponentially with respect to the number of nodes. Most commonly, these score-metrics are decomposable, which allows the score for a given DAG to be calculated as a sum (or

product) of the scores of the individual nodes. There are two different types of scoring functions to learn a structure from data [139]. The first type are ‘information theoretic scoring functions’, which are closely related to foundational work on complexity theory, randomness and the interpretation of probability [115]. The most well-known examples to these are Log likelihood, Bayesian Inference Criterion (BIC), Akaike Information Criterion (AIC), Minimum Message Length (MML) [140] and Minimum Description Length (MDL)[141]. Most of these information theoretic scores consist of a likelihood term and a penalty term that penalises complex network structures [133].

The second type are called ‘Bayesian scoring functions’, which start their search with a prior probability distribution over all possible causal models with the aim to find the model that maximises the posterior probability distribution given data [142]. The common examples of Bayesian scores (probability distributions) are K2, Bayesian Dirichlet (BD), Bayesian Dirichlet Equivalent (BDe) and Bayesian Dirichlet Equivalent with a uniform distribution (BDeu).

Most score-based structure learning algorithms operate on the (undirected) Markov-equivalent space [115], [118] rather than the DAG space of causal structures. These structures are referred to as partially directed acyclic graphs (PDAGs) [118]. Carrying out the search on PDAG space is motivated by the fact that two DAGs sharing the same skeleton and v-structures have identical independencies between variables. As a result, given only observational data, no unique BN can be distinguished since the scoring functions for Markov equivalent structures yield identical results.

In order to carry out score-based searches, various different algorithms have been proposed. The earliest algorithm that allowed learning discrete BN structures was K2 [143]. This algorithm takes advantage of a brute force greedy search and needs a specific node order to be input in order to reduce the search space. In principle, due to the exponentially increasing size of the search space, greedy search algorithms end up having

to make strong simplification assumptions to make the calculations relatively tractable. An alternative and more popular approach is using stochastic search on the available model space. Two prominent approaches commonly used for this purpose are Genetic Algorithms [144] and sampling methods such as Gibbs Sampling and Markov Chain Monte Carlo – in particular Metropolis Hastings- algorithms [118].

Apart from automated learning of the structure from data, manual construction of the structure of a BN is also a viable option, especially in domains where the causal relationships between different nodes are well-known by domain experts, such as medicine. Lucas et al. [114] report that many of the Bayesian Networks, developed for real life applications in biomedicine and healthcare have been constructed manually [127], [145]–[151].

In practice, both the automated learning and manual construction of DAGs have limitations. Manual construction in general is very time consuming and relies completely on the experts having complete domain knowledge. On the other hand, most automated learning algorithms are ineffective given small and noisy datasets. An alternative to both is hybrid approaches that combine automated learning with expert elicitation. This can simply be in the form of a score-based stochastic search method being initiated with an expert elicited DAG or the score metrics can be tailored to favour the structures that are in line with expert beliefs. Unfortunately, the utilisation of such hybrid methodologies is a relatively underexplored research area [122]. The most notable contribution in this topic is the Causal discovery via MML (CaMML) algorithm [115], [152], which can incorporate expert elicited hard and soft constraints into the structure learning process. We will discuss CaMML and other structure learning algorithms in more detail in Chapter 6.

2.4.2.3. Parameter Learning

After the DAG structure of the BN has been determined, it needs to be parameterised to specify the conditional probability tables that collectively form the joint probability distribution of the system given in equation (2.1). Here, we will focus on discrete probability distributions since the LUCADA dataset is made up almost exclusively of discrete variables. The probabilistic relation between a discrete node and its discrete parents (as given in Equation (2.1)) is encoded in the child variable's conditional probability table (CPT). While CPT's are flexible and intuitive representations for discrete local structures, the number of parameters that need to appear in a BN node's CPT is exponential in the number of parent variable combinations [115]. As a result, many structure learning algorithms employ 'max fan in' restrictions to define how many parent nodes a single node can have in order to keep the probability space tractable.

Although the parameters can be identified with expert elicitation, in cases where there is available data, learning the parameters from data via maximum likelihood estimation (MLE) [153], [154] is more common. As a matter of fact, it is so common that some AI researchers actually question whether BN's are actually 'Bayesian' since in most applications, their parameterisations are acquired from data frequencies [115].

Generally, discrete variables are modelled with binomial or multinomial distributions depending on their number of states. The simplest discrete probability distribution is the binomial distribution, which is used to model binomial variables such as patient gender in a population sample. As an example, with the binomial distribution, the likelihoods of probabilistic parameters, θ, given a sample dataset D can be given as:

$$L(\theta \mid D) = P(\mathrm{D}|\theta) = \prod_{i=1}^{N} \frac{N!}{D_i!(N-D_i)!} \theta^{D_i} (1-\theta)^{N-D_i} \qquad \textbf{(2.4)}$$

In the above equation, N is the sample size, and D_i is the i'th observation in the sample. In case we wish to use our prior belief in the estimation of Θ, using the Bayes' theorem [116], the estimation equation becomes:

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{\sum_{\theta_i} P(D|\theta_i)P(\theta_i)} \quad \textbf{(2.5)}$$

Compared to MLE and Maximum a Posteriori (MAP) estimations, Bayesian estimation is made more complex by the fact that we need to explicitly calculate the normaliser term in the denominator, also known as the marginal likelihood, or evidence. The derivation of a closed form for the probability of evidence becomes very challenging unless we are careful in the choice of the distribution $P(\Theta^f)$ to represent our prior belief. In BN problems, this prior distribution is most commonly represented in the form of a conjugate prior.

For a given likelihood function $P(D|\Theta)$, a prior $P(\Theta)$ is called a conjugate prior if the posterior $P(\Theta|D)$ has the same algebraic form as the prior. This naturally simplifies the computation of the posteriors, as the parameter estimation becomes a process of updating the prior's hyperparameters [155]. The conjugate prior for the binomial distribution is the beta distribution given as:

$$P(\theta \mid \alpha_1, \alpha_2) = \beta \cdot \theta^{(\alpha_1 - 1)} \cdot (1-\alpha_1)^{(\alpha_2 - 1)} \quad \textbf{(2.6)}$$

In (2.6), α_1, α_2 are termed the hyperparameters. Their selection is important since they specify the shape and characteristics of the prior distribution that represents our subjective beliefs. For the example given in (2.5), incorporating our prior belief into the estimation process results in:

$$P(D|\theta^f) \cdot P(\theta^f) = \beta \cdot \theta^{f+\alpha_1 - 1} \cdot (1-\alpha_1)^{(N-f)+\alpha_2 - 1} \quad \textbf{(2.7)}$$

As the reader will notice, this unnormalised posterior probability term is algebraically of the same form as (2.4) and (2.6). As a matter of fact, the hyperparameters of (2.6) are

integrated into the equation to pull the MLE towards the prior belief of the knowledge engineer. In (2.7), α_1, α_2 act as pseudo counts, playing the same role in subsequent learning as would an initial sample of size: $\alpha_1 + \alpha_2$. As a result, the hyperparameters can be thought of as a pseudo initial sample, and $\alpha_1 + \alpha_2$ is commonly referred as the "equivalent sample size". Accordingly, as the equivalent sample size is increased, the pull towards the prior belief becomes more prevalent. This overall process of parameter estimation with and without the integration of prior beliefs generalises directly to variables that have more than two states, i.e. multinomial variables, using another conjugate family of distributions, namely the Dirichlet family [156].

2.4.2.4. Inference

The joint probability distribution learned through the previous steps can be utilised to perform probabilistic inference on the BNs. Practically, when a set of evidence nodes are entered, Bayesian inference is used to update the probabilities of the query nodes, given the newly entered evidence. There are exact and approximate inference algorithms which propagate evidence input by the user to produce a new probability distribution over all nodes in a network [115].

The first BN inference algorithms were developed for tree structures [157], [158]. These mainly utilise belief propagation and message passing on a tree, which was then improved to apply to multiply connected networks [119], [159]. Among these, the junction tree algorithm is the most standard algorithm in which a BN is triangulated into a junction (join) tree structure and then a local message passing algorithm is run on this tree [116]. On the other hand, variable elimination is probably the simplest directed graph-only inference algorithm that works by exploiting the chain rule decomposition shown in (2.2), essentially pushing sums inside products. This was first introduced by Shachter et al. in 1986 in the context of decision networks [160].

Inference in BNs is a broad topic and apart from the two most commonly used algorithms, many other inference algorithms exist. Detailed reviews of the literature and technical details can be found in [115], [116], [119], [120].

Chapter 3 - LUCADA Dataset and Ontology

In this chapter, we introduce the LUCADA dataset, its underlying data model, and the LUCADA ontology, which will serve as the knowledge-base of our clinical decision support prototype in lung cancer care and as such should be capable of expressing all concepts and entities included in the LUCADA data model. In the first two sections, we discuss the structure, temporal characteristics, and the degree of completeness of the dataset, which will also set the stage for the following chapters where we discuss the design stages of the LUCADA Bayesian Network.

Following the treatment of the LUCADA data model, we discuss the various design steps in building the LUCADA ontology, which is a domain-specific standardised OWL-2 ontology. First, we summarise our initial attempts at ontology matching and automatic module extraction from SNOMED-CT. Following this, we describe our conceptualisation efforts and share the main heuristics adopted during our manual design of the LUCADA ontology. In addition, we present the process diagram through which we establish semantically accurate mappings between the LUCADA ontology and SNOMED-CT. We conclude the chapter by presenting our integrated LUCADA-SNOMED-CT ontology and by reiterating the major difficulties in ontology design and evaluation.

3.1. The LUCADA Dataset

In partnership with the Royal College of Physicians, the NHS Clinical Audit Support Unit runs the National Lung Cancer Audit (NLCA) on a contract from the Health Care Quality Improvement Partnership (HQIP). The aim of the audit is to improve the outcome for people diagnosed with lung cancer and mesothelioma. The data collected by NLCA is stored within the LUCADA dataset, which is a subset of the National Cancer Dataset plus a number of additional items. The motivation behind the audit and data collection is to gain

a better understanding of the care delivered during referral, diagnosis and treatment of lung cancer patients and relating these to patient outcomes [161].

Participation and data entry to the LUCADA dataset is mandated by NICE Guidelines. The size of the LUCADA dataset far surpasses any other lung cancer patient dataset worldwide. Aiming to gain access to the LUCADA database, we first contacted the clinical lead of the NLCA, Dr Michael Peake, and the project manager Dr Roz Stanley in January 2011. Gaining access to an anonymised version of the database took approximately 5 months from the drafting of the research proposal to the eventual signing of the first data sharing agreement (DSA) between the NHS and the University of Oxford.

The DSA allowed all non-sensitive LUCADA patient fields to be made available to the university in order to carry out research in biomedical engineering fields of clinical decision support and machine learning in order to build a software tool to assist clinicians in arriving at informed, timely, safe and effective decisions in lung cancer care. As a result, all sensitive patient identifiers such as postcode, birth date, NHS number and death date were excluded from the anonymised data transfer. In addition, data from trusts submitting less than 10 cases and data with inconsistent diagnosis and death dates were not included.

We received the first batch of anonymised patient data on 1 June 2011, which consisted of 115,712 English patients with dates first seen from early 2004 to the end of 2009. This first batch of data was very useful in attaining a better understanding of the lung cancer domain and the variables that are vital in treatment plan selection. We have also used this dataset to run our initial studies in determining the different ways that probabilistic inference can help uncover clinically meaningful information from the LUCADA dataset [37]. One suboptimal feature of this first dataset was the data quality and completeness levels. The NLCA annual reports from 2007 to 2011 all indicate that the quality and completeness of data submitted by trusts improve with each passing year. As an example, the “pre-

treatment TNM stage" field completeness rose from 47 % in 2006 to 59 % in 2007, and then to 71 % in 2010.

In order to make use of the data with the highest quality and completeness levels, in October 2012 we requested the renewal of the existing DSA in order to include the most recent patients. The data transfer of this second batch of patients occurred on 21 November 2012, including 126,896 English patients with dates first seen from 2006 to the end of 2010. Following the terms agreed upon in the first DSA, this data transfer included 98 LUCADA fields. The following section introduces the structure, characteristics and the levels of field completeness of the LUCADA dataset.

3.2. The LUCADA Data Model

A complete list of all LUCADA data fields, along with their full definitions and the list of values they can take, is given in the LUCADA Data Manual document [16] available on the web. This section is intended to outline all the different data fields but will mainly focus on those that have a direct clinical influence on treatment selection and prognosis.

In general, LUCADA data fields are organised into six sections, namely: Patient Referral, Care Plan / MDT, Key Investigations, Nursing Care, Treatment, and Outcome. All of these data fields store categorical data, except for three: "*Age at time of diagnosis*", "*FEV1 Percentage*" and "*FEV1 Absolute amount*". Among these three, the "*Age at time of diagnosis*" field is ordinal with integer values ranging from 3 to 107; "*FEV1 Percentage*" is again ordinal with integer values ranging from 1 to 150; and finally "*FEV1 Absolute amount*" is continuous, taking on real values between 0.1 and 10.

As the data is collected for audit purposes, LUCADA includes a lot of administrative fields in addition to the clinically more useful patient and disease-specific ones. These mainly comprise of dates or site codes of individual treatments and diagnostic scans. There are also a couple of nursing care fields that fall under the 'administrative' category.

While these fields are obviously valuable for keeping track of the treatment time scales and the availability of resources like nursing care and specific diagnostic modalities in different hospitals, for our purposes we are mainly interested in fields that directly influence the clinical decision making process and the patient outcomes. It has been noted that the quality of care given in different cancer centres varies and so information about the location where the patient has been diagnosed or treated should make a difference to treatment selection and survival. However, despite being important, a comparison of the performances of different cancer centres falls out of the scope of this research.

The rest of this thesis focuses on the non-administrative fields that directly affect both treatment selection decisions and treatment outcomes. These fields were selected by the NLCA clinical lead, Dr Michael Peake, and our clinical collaborators in the Oxford University Hospitals: Prof Fergus Gleeson and Dr Donald Tse. As already set out in Chapter 1, within the context of clinical decision support, the two significant time points in the patient journey are treatment selection and treatment outcomes. In light of these observations, the clinical (non-administrative) LUCADA data fields will be introduced with respect to their temporal order in the patient journey, namely: 'Pre-Treatment', 'Treatment', and 'Outcome' fields.

3.2.1. Pre-treatment fields

Pre-treatment fields are those that are required before a treatment decision is made. A complete list is given in Table 3.1, which includes 1) demographic fields such as "*Sex*" and "*Age at time of diagnosis*"; 2) fields that correspond to the physical well-being of the patient, such as "*Performance Status*", "*FEV1 values*" and comorbidities (colour-coded in blue) and finally 3) fields that detail the disease type, location and stage.

Data Field	Type	Completeness %
Sex	Binary F/M)	100.00
Age at time of diagnosis	Integer	100.00
Staging identifier	Binary (6/7)	99.98
Primary diagnosis	Categorical	96.97
Tumour laterality	Categorical	93.37
T Category (pre-treatment)	Categorical	70.25
N Category (pre-treatment)	Categorical	66.85
M Category (pre-treatment)	Categorical	67.53
TNM category (pre-treatment)	Categorical	71.77
Histology (SNOMED)	Categorical	65.78
Site-specific staging classification	Categorical	7.03
Dementia/cerebrovascular disease	Binary (Y/N)	6.44
Cardiovascular disease	Binary (Y/N)	8.10
Renal failure	Binary (Y/N)	5.69
Other malignancy	Binary (Y/N)	7.25
Severe weight loss	Binary (Y/N)	6.33
Other significant co-morbidity	Binary (Y/N)	10.41
FEV1 Absolute amount	Real	30.54
FEV1 Percentage	Integer	24.88
Performance Status (adult)	Categorical	86.64

Table 3.1: The pre-treatment data fields in the LUCADA dataset

Among the disease-specific fields, "*Primary Diagnosis*" records the ICD-10 code [162] that best describes the anatomical site of the primary cancer. The "*TNM category (pre-treatment)*" field stores the definitive pre-treatment stage of the cancer, with the constituent T, N and M staging information reported in separate fields as well. Figure 3.1 gives the TNM staging distributions of the LUCADA patient population. In line with our statement in Chapter 1, the figure shows that only 13% of the lung cancer patients in England are diagnosed at an early stage (IA-IIB), which -as we will observe in Section 3.2.3- has a rather negative impact on the survival rates.

The "*Staging Identifier*" field indicates whether version 6 or 7 of the TNM staging has been used in the staging of the patient's disease. As explained in Chapter 1, TNM version 7 was only introduced in 2009. Since our dataset spans 2006 to 2010, the percentage of the patients staged using version 6 (88%) is much higher than those staged with TNM version 7 (12%).

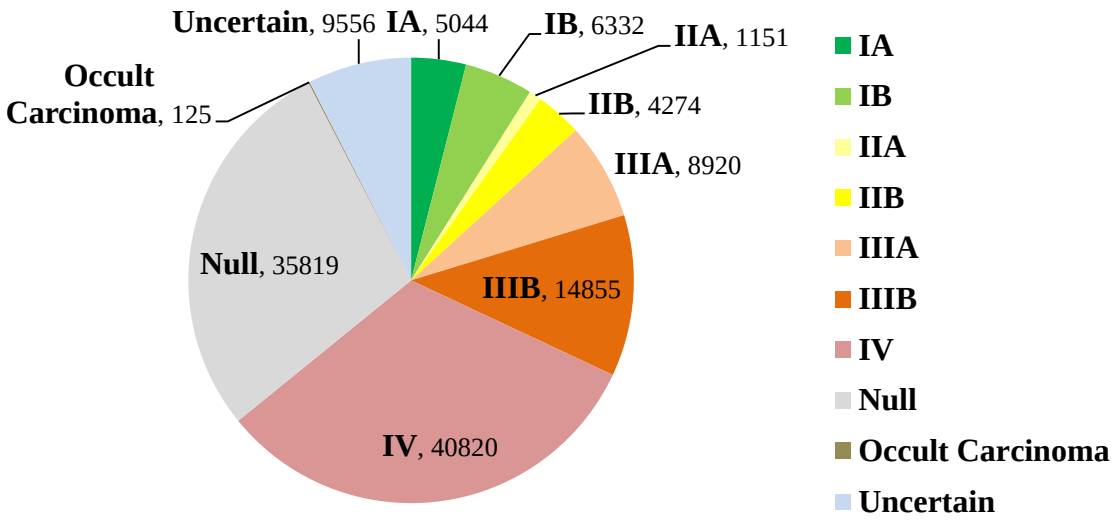

Figure 3.1: The TNM staging distributions of the LUCADA patients. The data labels read as 'TNM Stage, Number of Patients**'.**

The "*Histology (SNOMED)*" field records the SNOMED code of the histology type of the primary tumour. This field can take on 19 different values, ranging from subtypes of Non-small cell lung cancer (NSCLC) to Small cell lung cancer (SCLC) and Mesothelioma. As already stated in Chapter 1, Mesothelioma has been excluded from the scope of this thesis. After excluding the mesothelioma histology types, we rearranged the 15 remaining histology types into eight different groups as advised by our expert panel. These 8 histology types are shown in Figure 3.2, which highlights the fact that for a significant portion of the population (33%), no histology data has been entered. Among the cases which have histology data, NSCLC cases altogether approximate to 78.5%, SCLC cases to 16% and Other Histology type cases to 5.5% of all population.

Among the data fields that define the physical well-being of the patient, "*Performance Status (adult)*" is the most notable. This field uses the World Health Organisation (WHO) scale between 0 and 4 to indicate the overall fitness level of a patient.

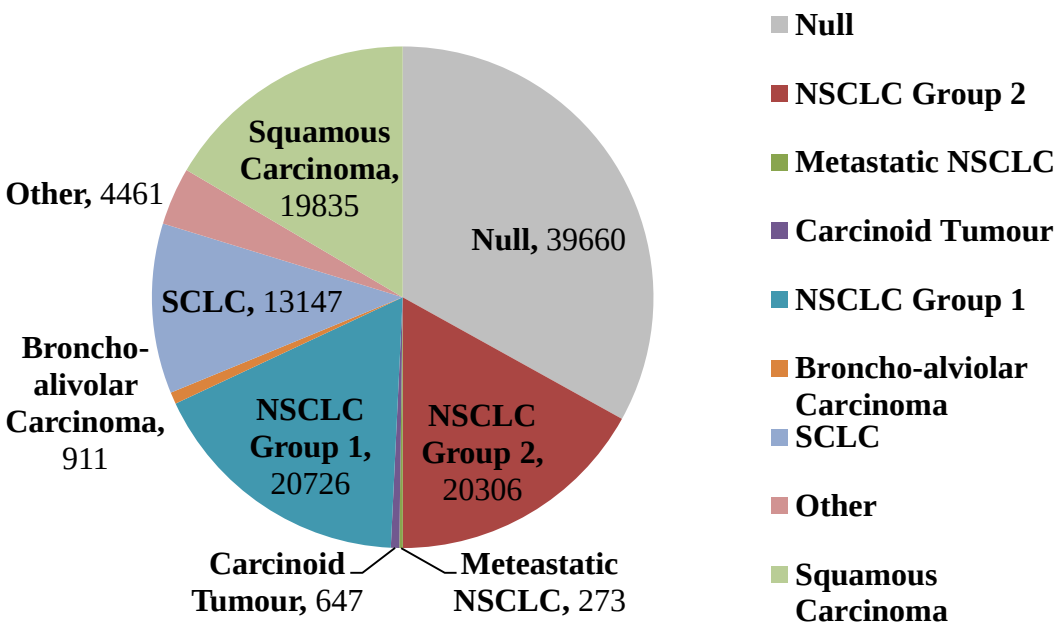

Figure 3.2: The main histology types of the LUCADA patients. The data labels read as 'Histology Type, Number of Patients**'.**

Figure 3.3 gives the performance status distribution of the LUCADA population. The peculiar thing with this data field is that, apart from patient entries with missing (null) performance status, there are also some which are not null but filled in with the 'Not recorded' option. When these entries are added to the Null ones, the overall completeness level of the "*Performance Status (adult)*" field reduces to 72%.

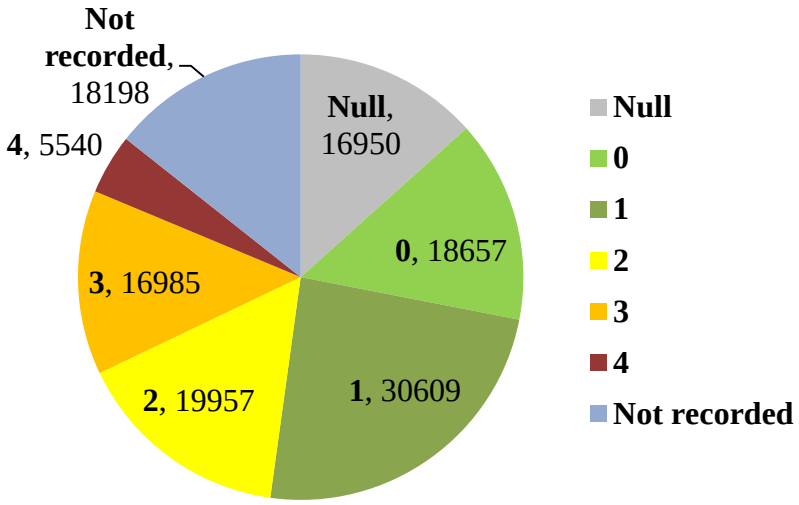

Figure 3.3: The WHO performance status distribution of the of the LUCADA patients. The data labels read as 'WHO Performance Status, Number of Patients**'.**

In addition to the performance status of the patient, LUCADA includes 6 binary comorbidity fields, which are colour-coded in light blue in Table 3.1, to indicate the presence of the most commonly encountered comorbidities in lung cancer patients. However, as can be seen in Table 3.1, the completeness levels of these fields are significantly low. In fact, the NLCA clinical lead, Dr Michael Peake, states that the comorbidity fields in LUCADA are among the ones that have the most room for improvement in terms of data completeness. The other two data fields in the table, which have low completeness levels, are "*FEV1 Absolute amount*" and "*FEV1 Percentage*". As already explained in Chapter 1, these two fields are associated with lung capacity and fitness, which become important deciding factors especially for patients being considered for surgery.

3.2.2. Treatment fields

As discussed in Chapter 1, the LUCADA data model has various fields that hold information on the suggested cancer treatment plan and the details of the constituent treatments. The overall treatment plan for a patient is stored in the "*Suggested Cancer Treatment Plan*" field, which can take one of the 11 possible options listed in Table 3.2.

Code	Name
1	Surgery
2	Teletherapy / Radiotherapy
3	Chemotherapy
4	Brachytherapy
5	Palliative care
6	Active Monitoring
7	Sequential chemotherapy and radiotherapy
8	Concurrent chemotherapy and radiotherapy
9	Induction chemotherapy to downstage before surgery
10	Neo-adjuvant chemotherapy and surgery
11	Surgery followed by adjuvant chemotherapy

Table 3.2: The 11 available treatment plan options in LUCADA

The "*Suggested Cancer Treatment Plan*" field is empty for around 14% of the patients. Following advice from our expert panel, we removed the Brachytherapy treatment plan since it was not deemed to be a common treatment plan type for lung cancer and accordingly had only been prescribed for less than 100 patients in the entire dataset. Figure 3.4 below outlines the distribution of the entire LUCADA patient population with respect to their recorded treatment plans. As can be observed, the five most common treatment plans in descending order are: 'Palliative Care (5)', 'Chemotherapy (3)', 'Radiotherapy (2)', 'Surgery (1)' and 'Active Monitoring (6)'. An important thing to note is that the proportion of the multiple modality treatment plans - with codes 7 to 11 - is relatively small. One of the reasons for this small ratio of multiple modality treatment plans is mainly data accuracy, as will be discussed in Section 3.2.4. Furthermore, the clinical evidence for 'Neo-adjuvant chemotherapy and Surgery (10)' and 'Surgery followed by adjuvant chemotherapy plans (11)' is still not very strong and the patients who have been given these treatment plans are those who are in clinical trials.

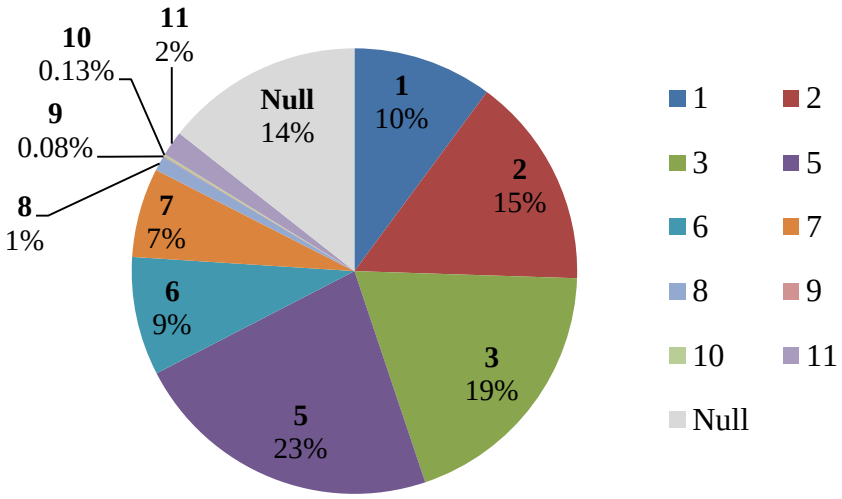

Figure 3.4: Suggested cancer treatment plan distributions in the LUCADA dataset. The data labels read as 'Treatment Plan Type, Number of Patients**'.**

In order to complement the information on treatment plan distributions in Figure 3.4, we also investigated the 1-year survival rates per each treatment plan type. Figure 3.5 gives the

survival percentages stratified with respect to treatment plan types. In the figure, the expected value of 1-yr survival across the database (32.6%) is marked with a blue line.

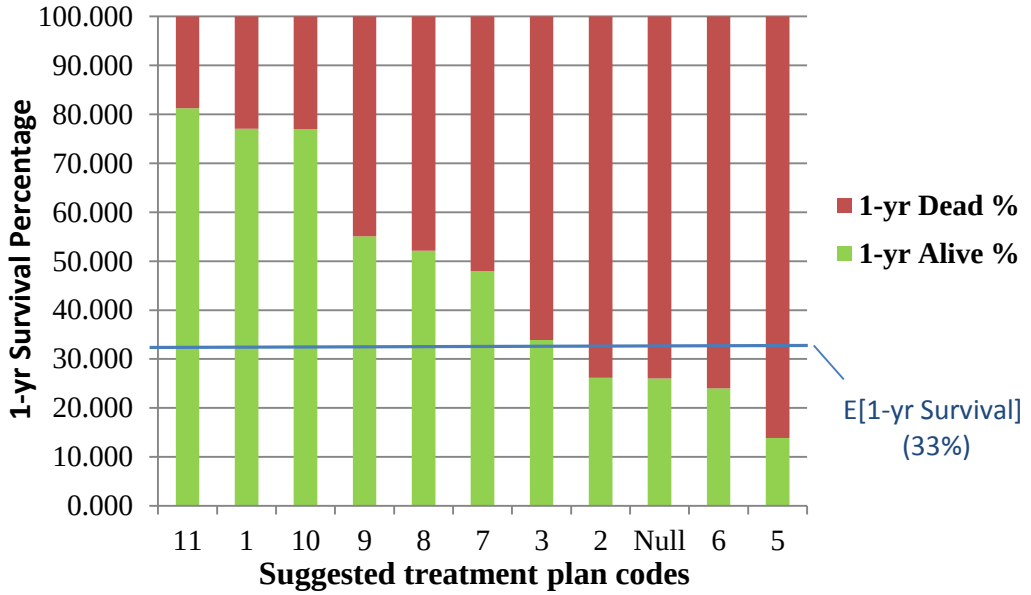

Figure 3.5: Survival percentages with respect to suggested cancer treatment plans in the LUCADA dataset.

Figure 3.5 reveals a clear pattern between the 1-year survival rates of different treatment plan types. As can be seen in the figure, the four treatment plans with the highest survival rates (11, 1, 10 and 9) are those that involve surgery. Following these, the columns in the middle with worse 1-year survival rates belong to non-surgical curative treatment plans as: Chemotherapy, Radiotherapy or a combination of the two (3, 2, 8 and 7). Finally, the last two columns with the worst survival rates are Active Monitoring (6) and Palliative Care (5), which is unsurprising since they are both non-curative treatments.

The other two data fields that identify the treatment plan are "*Cancer Care Plan Intent*" and "*Planned Treatment type*". The former records the intention of the suggested treatment plan, which can be 'Curative', 'Palliative', 'Supportive' or 'No Anti-cancer Treatment' and the latter captures whether the suggested treatment plan is of single or multiple-modality.

In addition to the more general treatment plan related fields, LUCADA also contains treatment-specific fields that provide some (limited) detail regarding the individual treatments given such as: "*Chemotherapy treatment given*", "*Radiotherapy treatment*

given" and "*Palliative care provider type*". Unfortunately, these data fields are not very detailed and for the time being vital information such as the exact chemotherapy drug regimen or radiotherapy dosage and frequency are not recorded in LUCADA. The NLCA aims to start collecting these data in the future as part of the audit in order to improve the granularity of information regarding the treatments given to patients. A complete list of the treatment-specific data fields is given in Table 3.3.

Data Field	Type	Completeness %
T Category (pathological)	Categorical	12.92
N Category (pathological)	Categorical	8.22
M Category (pathological)	Categorical	7.63
TNM Category (pathological)	Categorical	16.76
Histology (SNOMED) (post)	Categorical	10.37
Post treatment: Site specific classification	Categorical	0.43
Primary procedure (OPCS)	Categorical	10.56
Excision margin	Categorical	4.53
Radiotherapy anatomical site	Categorical	17.12
Confirm prophylactic cranial irradiation	Binary (Y/N)	43.86
Radiotherapy treatment given	Categorical	21.88
Chemotherapy treatment given	Categorical	21.66

Table 3.3: The treatment-specific data fields in the LUCADA dataset

The reader may note that the completeness levels of these data fields are relatively low. This is due to the fact that these treatment specific fields become applicable only if the "*Suggested Cancer Treatment Plan*" for that patient includes any of the individual treatment types corresponding to one or more of these fields. For instance, the data fields colour-coded in green in Table 3.3 only become relevant if the patient has been prescribed a "*Suggested Cancer Treatment Plan*" option that involves surgery – namely: 1, 9, 10 or 11. In other treatment plans that do not involve surgery, no surgery-related fields are applicable.

3.2.3. Outcome fields

The DSA between the NHS and Oxford University allowed seven outcome-related data fields to be included in the data transfer. Among these seven, "*Dead/Alive*" and "*Survival*

(days)" are derived fields that originally did not exist in LUCADA. Following our request, these two fields have been included in the data transfer as non-sensitive survival indicators. The binary "*Dead/Alive*" field is derived from the "*Death Date*" field in LUCADA, which is taken from the Office for National Statistics (ONS). The "*Survival (days)*" field is calculated as the difference between "*Death Date*" and "*Diagnosis Date*" of the patient if the patient is dead. If the patient is still alive, it is calculated as the difference between "(*Data) Extraction Date*" and "*Diagnosis Date*".

All outcome-related LUCADA fields are listed in Table 3.4. In addition to the two derived survival fields, LUCADA includes five data fields that provide additional information on treatment outcome. However, as can be seen from this table, the completeness levels of these treatment outcome fields are significantly low.

Data Field	Type	Completeness %
Dead/Alive*	Binary (D/A)	100.00
Survival (days)*	Integer	97.14
Death – was death related to treatment	Binary (Y/N)	22.85
Treatment morbidity type (cancer)	Categorical	0.88
Original treatment plan carried out	Binary (Y/N)	36.61
Original treatment plan failure reason	Categorical	3.95
Reason patient did not receive 1st treatment choice	Categorical	13.38

Table 3.4: The outcome-related data fields in the LUCADA dataset. The derived survival fields are marked with an asterisk (*).

The most common clinical outcome indicator reported in trial results (and therefore adopted by guideline rules) in cancer care is 'disease-free survival'. In particular, 5-year survival rate is the most commonly used cut-off point to measure disease-free survival. However, as mentioned earlier, the LUCADA dataset only includes patients who have been diagnosed between 2006 and 2010 and does not contain many patient data on 5-year survival as of yet. As a result, we chose to use 1-year survival as a surrogate outcome measure. This choice was reinforced by our expert panel and literature evidence, which reports that almost all improvement in lung cancer survival is attributable to an increase in 1-year survival [163]. In their lung cancer survival benchmark study, Holmberg et al [164]

also report that "*the difference in excess risk of dying between different countries was predominantly confined to the first year of follow-up after diagnosis*". As a result, we derived the 1-year survival field from the "Dead/Alive" and "Survival (Days)" fields. Among all patients in the dataset, the expected value of 1-year survival rate was approximately 33%.

Given its size and breadth, even becoming familiar with the LUCADA dataset has been arduous. In the next section, we will discuss the major steps in pre-processing and cleaning the dataset.

3.2.4. Data Cleaning

One inherent limitation of the LUCADA database is that the data are entered manually in cancer centres and are therefore error-prone. Some of these errors are implicit and cannot be corrected with the information available in the observed data. For instance, if the "Primary Procedure" type given for a patient has been erroneously recorded as 'Pneumonectomy' whereas the actual treatment given was 'Lobectomy', we have no way to recognise and fix this. Fortunately in some cases, the data entry errors are obvious and can be treated with a pre-processing step. Examples of such obvious errors that can be treated by pre-processing are:

1) Inconsistent Care Plan Intent Entries: In some cases, the intent of the treatment plan can be deduced from the definition of the "Suggested Cancer Treatment Plan" value. For instance, we can safely assume a data entry error if a 'Palliative Care Plan' has a 'Curative' intent or a 'Surgery Plan' has "No anti-cancer treatment" intent. We have manually corrected such obvious errors in the dataset.
2) Inconsistent "Planned Treatment Type" Entries: Similarly, whether or not a treatment plan type is multiple modality can be deduced from its name and therefore we can assume a 'Sequential Chemoradiotherapy Plan' has to be of type 'Multiple Modality'.

We fixed any inconsistencies of this type before further carrying out further analysis on the dataset.

3) Inconsistent "Suggested Cancer Treatment Plan" Entries: The most substantial inconsistencies occur when the treatment related fields (listed in Table 3.3) collectively imply a different treatment plan than the one recorded under the "Suggested Cancer Treatment Plan" field. We have observed that these inconsistencies arise for three different reasons:

i) Data Entry Errors: Sometimes, the "Suggested Cancer Treatment Plan" field is filled in with a completely different value to the individual treatment type(s) recorded. An example is a patient entry with "*Suggested Cancer Treatment Plan*" entered as 'Chemotherapy Plan', and with no Chemotherapy related fields filled in, while all Radiotherapy related fields are complete. We treated all such cases manually.

ii) As given in Figure 3.4, approximately 38,000 patients in the dataset do not have a specific "Suggested Cancer Treatment Plan" recorded. However, a substantial proportion of these cases actually have treatment specific details that provide sufficient evidence that a particular treatment plan has actually been given to the patient. For these cases, based on the available evidence in the dataset, we manually filled in the appropriate "Suggested Cancer Treatment Plan" state. As an example, there are 3423 patient records in the dataset, which do not have a treatment plan recorded but have their "Care Plan Intent" filled in as 'Palliative' and with an existing "Palliative Care Provider Type" or "Date of Interventional Treatment Given". It is safe to assume that these patients have indeed received Palliative Care, which was not recorded under "Suggested Cancer Treatment Plan" so we changed their treatment plans from being empty to palliative care.

iii) Incomplete Multiple Modality Errors: The clinical lead of the NLCA reports that while the "*Suggested Cancer Treatment Plan*" is more than 80% accurate in terms of

recording the first treatment given, it is less reliable for multiple modality treatment plans. We observed many patient entries that have been given more than one treatment type but recorded as a single modality treatment plan. In most such cases, the "*Suggested Cancer Treatment Plan*" reflects the first treatment given. For instance, a "*Chemotherapy Plan*" patient, who has subsequently been given radiotherapy without updating the "*Suggested Cancer Treatment Plan*", field to 'Sequential Chemoradiotherapy' falls under this category. Similarly, a "Surgery Plan" patient, who has been also been given Chemotherapy after their surgery, without the "*Suggested Cancer Treatment Plan*" being updated to "Adjuvant Chemotherapy Plan" is another example to such inconsistencies.

For such cases, since the information conveyed by the treatment-related fields is more detailed, whenever there was an inconsistency between the treatment fields and the suggested treatment plan, we have corrected the "*Suggested Cancer Treatment Plan*" to be consistent with the treatment related fields. Table 3.5 lists the treatment plan corrections made in the database. As the reader may notice, the highest number of corrections was made for patients whose recorded "Chemotherapy" treatment plans were corrected to "Sequential Chemoradiotherapy" in the light of the evidence provided by the treatment detail fields.

Original Treatment Plan	**Corrected Treatment Plan**	**Number**
(3) Chemotherapy	(8) Concurrent Chemoradiotherapy	84
(3) Chemotherapy	(7) Sequential Chemoradiotherapy	2918
(3) Chemotherapy	(11) Adjuvant Chemotherapy	140
(3) Chemotherapy	(10) Neo-adjuvant Chemotherapy	36
(2) Radiotherapy	(8) Concurrent Chemoradiotherapy	42
(2) Radiotherapy	(7) Sequential Chemoradiotherapy	855
(1) Surgery	(11) Adjuvant Chemotherapy	957
(1) Surgery	(10) Neo-adjuvant Chemotherapy	28
SUM:		**5060**

Table 3.5: The number of manual Suggested Cancer Treatment Plan field corrections we have made in the LUCADA dataset.

3.3. LUCADA Ontology

As discussed in the previous section, the LUCADA data model is represented in a flat and tabular structure that includes a list of data items together with certain values they can assume. In Chapter 2, we discussed that modelling domain knowledge in the form of ontologies has noticeable benefits, such as the ability to easily interface with other applications that use the same reference ontologies and the formal models of semantic reasoning they support. In particular, we dwelt on the practical advantages of formalising this knowledge using OWL-2 [80].

In order to take advantage of all these benefits, we modelled the LUCADA data model as an OWL-2 ontology that would contain all entities (data items and the values they can take) in LUCADA, therefore enabling a semantically accurate mapping of patient entries between the database and the ontology.

3.3.1. Adopting SNOMED-CT

In many disciplines, due to the lack of generalisable reference ontologies, researchers are generally attracted to designing their own ontologies that represent the domain from their idiosyncratic viewpoint. This results in a myriad of different software applications, operating on similar domains but 'speaking different languages', which ultimately confines the use of their technology and knowledge to their home institutions. This heterogeneity greatly hinders interoperability and sharing medical knowledge across computational resources. During the literature review of Computer Interpretable Guideline (CIG) formalisms in Chapter 2, we have stressed the importance of interoperability, in particular for clinical decision support (CDS) applications.

In the domain of medicine, there are many reference ontologies that one can adopt in their ontological applications. In addition to facilitating interoperability, using a reference ontology also reduces development costs. We listed the most common reference ontologies in Section 2.2.5, reporting that among others "*SNOMED-CT has been approved by the*

Information Standards Board as the full fundamental standard for all medical information applications within the NHS". In addition to its geographical relevance, SNOMED-CT is an HL7-endorsed medical nomenclature standard and is deemed to be the *lingua franca* of medicine by medical informatics experts globally [165].

Hence, we chose SNOMED-CT as the reference ontology to use in the Lung Cancer Assistant (LCA). However, due to its enormous size, even browsing the SNOMED-CT is a computationally intensive task, let alone reasoning over. In order to address this problem and exclude 'redundant' concepts, which are not immediately relevant to the LUCADA data model, we opted to extract a domain specific subset (module) of SNOMED-CT to act as the LCA's knowledge base.

3.3.2. Automatic Module Extraction Attempt

The design, reuse, and integration of ontologies are complex tasks that require software tools and methodologies which assist the ontology engineers during these processes and minimise the introduction of inconsistencies [166]. The major steps we had to accomplish for extracting a SNOMED-CT module based on the LUCADA data model were: 1) Conceptualisation, which involves the formalisation of all concepts and properties that collectively represent our domain knowledge. 2) Ontology Matching, which consists of finding accurate mappings between our conceptualisation and the target ontology, in our case SNOMED-CT. 3) Module Extraction, which includes extracting a subset of the target ontology that preserves all semantic knowledge provided by the mapped concepts that are uncovered during ontology matching. These are discussed in turn.

3.3.2.1. Conceptualisation

Our anonymised LUCADA patient dataset is stored physically within a PostgreSQL [167] relational database table with every column corresponding to a patient field and the rows representing different patient entries. While being very efficient data storage tools,

relational databases do not provide explicit and formal semantics for the data they store [168]. Therefore, in order to formally represent the lung cancer care domain knowledge as an ontology, we first needed to create a 'conceptualisation' [169] by establishing direct correspondences between the database schema and our proposed LUCADA ontology.

In cases where the source database has a rich schema with many tables, relations and constraints, this conceptualisation can -to a certain extent- be 'reverse-engineered' from the database schema [170]. However, in our case there was no such distinctive schema. Therefore, we decided to base our conceptualisation on the column names (LUCADA data field names) and the values they can take (LUCADA data values) in our database table. For this purpose, we had to include entities that corresponded to both. As an example, we needed to represent both the data item "Tumour Laterality", and the values it can take ('Right', 'Left', 'Bilateral', etc.) in our ontology.

We initially attempted to automatically generate our list of concepts by running SQL queries that returned all column names and values within the database. However, this approach did not succeed due to the fact that in the dataset values are stored as codes rather than as character values that captured the semantics. For instance, the values of the "Tumour Laterality" column are stored as 'R' and 'L', standing for 'Right', 'Left'.

We therefore had to manually retrieve all data items and data values of LUCADA from the LUCADA data manual document [16] in order to create the 'LUCADA concepts list' that included 98 data item names and 144 data item values, together summing up to 242 entities.

3.3.2.2. Ontology Matching

The second step was to find the correct mappings of the 242 entities in our LUCADA concepts lists to SNOMED-CT. However, given the size of SNOMED-CT, manual discovery of these mappings was an arduous task that would, if possible, be better

automated. Therefore, we investigated various tools that could aid us in the process. In computer science, such tools are named ‘ontology matchers’. They are developed with the intention to match entities (classes, properties, individuals) between disparate ontologies and suggest mappings with the overall aim to overcome the problem of semantic heterogeneity.

The development of efficient and accurate ontology matching tools has recently become a popular research topic [171], [172]. Most ontology matchers operate on OWL and OWL-2, suggesting candidate mappings primarily based on lexical similarities between class names of different ontologies. Many of them can also carry out semantic verifications in order to reinforce the mapping suggestions. The semantic verification process is performed based on the hierarchical structures and the logical constraints (such as ‘disjointness’) of the ontologies. Unfortunately, as we found out at the end of our fully automated ontology matching attempt, in the absence of a similar hierarchical structure or very similar concept and property names, ontology matchers cannot perform well.

ASMOV [173], GOMMA [174], LogMap [175], SAMBO [176], Falcon [177] and KOSIMap [178] are among the commonly used ontology matching tools. However, most of these tools do not scale to work with really large ontologies such as SNOMED-CT, while others cannot make use of semantic similarities or provide interactivity during the mapping process so that the developer can give feedback to the system to help improve performance.

LogMap2 [179] has recently been developed in the Oxford Computer Laboratory as a much improved version of its predecessor: LogMap. It is currently the only ontology matcher that simultaneously provides scalability, user interaction and semantic verification. In collaboration with the research group of Professor Ian Horrocks, we attempted to use this state-of-the-art tool for extracting a “LUCADA module” from SNOMED-CT.

In order to make use of LogMap2, we needed to represent our conceptualisation in OWL-2 format, which would serve as our input ontology to be matched with SNOMED-CT. For this purpose, we made use of the OWL API and wrote a Java programme that accepts the entities in our LUCADA concepts list in comma separated (.csv) format and outputs a 'flat' OWL-2 ontology, containing all 242 concepts as OWL-2 classes without any hierarchical structure.

To gain access to our (foreign) target ontology: SNOMED-CT, we registered with the UK Terminology Centre (UKTC) of the NHS, which is the responsible body for managing SNOMED-CT in the UK. Following our registration, we downloaded the most up-to-date version of SNOMED-CT. SNOMED-CT is not distributed in OWL format but is stored as a collection of different tables that contain the list of concepts, attributes and relationships separately. However, the SNOMED-CT distribution comes with a "Description Logic Transform" script written in the Perl language, which we made use of to convert these collection of tables into an OWL ontology. The details of this transformation can be found in the SNOMED-CT Technical Implementation Guide, Section 4 [180].

Once we had our automatically created flat LUCADA ontology and an OWL version of the SNOMED-CT's July 2011 release, we ran LogMap-2 to perform an ontological mapping between the two. In order to assess the effect of LogMap-2's user interaction functionality, we repeated our experiment with and without user interactivity and measured the recall – the ratio of retrieved classes that are relevant- and the accuracy – the ratio of semantically accurate mappings- of the retrieved classes. The results of these experiments are as given in Table 3.7. As can be seen, out of the 242 classes in the automatically created LUCADA Ontology, LogMap-2 managed a recall rate of 0.28 (67/242) without user interaction. With user interaction, LogMap-2 initially reported 54 reliable mappings and 24 candidate mappings that needed user interaction to be manually verified. After our feedback, LogMap-2 recommended 67 mappings, with a recall rate of 0.28 again.

	# LUCADA Classes	# Output Mappings	# Correct Mappings
Without User Interaction	242	67	53
With user Interaction	242	67	56

Table 3.6: The results of ontology matching between the automatically created LUCADA ontology and the SNOMED-CT.

In order to verify their accuracy, we manually assessed the recommended mappings for both experiments and found that in the initial run with no user interactivity, the accuracy of the recommended mappings was 0.79 (53/67) and in the second run -where the system asked for our feedback- the accuracy of the mappings was (56/67) 0.84. The results indicated that while the user interaction resulted in the same recall rates, it helped improve the accuracy of the mappings. However, the significantly low recall rate constituted a major problem.

As mentioned previously, ontology matching still relies heavily on heuristics such as lexical and semantic similarities and is inevitably error-prone [79]. The low rate of recall (0.28) for both experiments can be attributed to two major factors. First, the automatically created LUCADA ontology included class names directly copied from the LUCADA data manual. When we analysed the output mappings, we saw that the average lexical similarity scores for the mapping pairs were significantly low, which indicates that the string matching function of LogMap-2 could not find enough lexical similarities between the unprocessed names in our LUCADA concepts list with the corresponding SNOMED-CT classes.

Second, the flat structure of the automatically generated LUCADA ontology meant there was no hierarchical indexation within the ontology and therefore semantic similarity measures could not be utilised at all by the ontology matching tool. In light of these results, we concluded that carrying out a completely automated ontology matching was not yet feasible even with a sophisticated and state-of-the-art tool as LogMap-2.

3.3.2.3. Module extraction

Module extraction is a key task in the partial reuse of ontologies, which usually involves extracting meaningful and minimal domain-specific subsets from large reference ontologies. There has been growing interest in this field in the recent years [181]–[183]. In 2007, Grau et al. showed that the problem of determining whether a subset of an ontology is a module for a given vocabulary is undecidable for the Description Logics underlying OWL-DL [166], [183]. Furthermore, they proposed a DL-based modularisation algorithm that made use of logical notions as 'conservative extension', 'safety' and 'locality' to provide an approximate solution to this undecidable problem. Their algorithm was implemented in 2007 by Jimenez-Ruiz et al. as a software tool named Locality Module Extractor [184].

Our final module extraction step was aimed at extracting a fragment of SNOMED-CT that guarantees to completely capture the meaning of all mapped concepts uncovered during the ontology matching step. More formally, we needed to isolate a minimal SNOMED-CT module that includes all related axioms to our mapped entities. Thus, in practice, when answering domain specific queries, importing this minimal module should give us exactly the same answers as if we had imported the whole of SNOMED-CT [166]. Obviously, the added value of this effort would be more efficient reasoning due to the decreased size of the ontology module.

However, the low recall rates of the previous ontology matching step for the automatically created flat LUCADA ontology meant that we did not yet have a satisfactory set of mapped concepts for producing a reliable SNOMED-CT module that fully represents the LUCADA data model. Even if we could make use of the few accurate concept mappings as a starting point for our domain specific LUCADA ontology, the a-posteriori manual curation of these mappings, and the manual addition of the classes that LogMap-2 failed to

match would still require a substantial amount of time and effort. Therefore we set out to manually build the LUCADA ontology to be mapped to SNOMED-CT subsequently.

3.3.3. Designing the LUCADA Ontology

In order to manually extract a LUCADA module from SNOMED-CT, we needed to select those SNOMED-CT concepts and relationships which were necessary and sufficient to express our domain conceptualisation. We wanted the resulting module to be compact since it was going to be deployed as the knowledge-base of a clinical decision support tool and therefore scalability and real-time inference performance were major design concerns.

The discovery of the appropriate SNOMED-CT concepts and relationships required a thorough investigation of the SNOMED-CT, most of the time forcing us to manually search for suitable SNOMED-CT class and property candidates that could represent the entities in our manually extracted LUCADA concepts list. As explained in Chapter 2, SNOMED-CT contains more than 344,000 active concepts, which did not make this task any easier. Fortunately, there are purpose-built SNOMED-CT browsers for carrying out relatively quick and efficient searches within this reference ontology. We made use of a popular and freely available SNOMED-CT browser software named CliniClue® Xplore [185] for this purpose.

Some entries in the LUCADA entities list, such as Histology and disease types, were straight-forward to map to SNOMED-CT concepts since they were, from the outset, adopted directly from the ICD and SNOMED codes by the developers of LUCADA. However, some data items did not have one to one mappings with a SNOMED-CT concept. In such cases, the data items were modelled as post-coordinated concepts making use of the atomic concepts and attributes available in SNOMED-CT.

For instance, there was no concept in SNOMED-CT that semantically matched the description of the ‘Severe Weight Loss’ data item, which is one of the comorbidity types in

LUCADA. However, being the most comprehensive medical ontology, SNOMED CT has "*Weight Loss Finding*" as a clinical finding concept, "*Severity*" as an object property, and "*Severe*" as a special atomic mapping value concept. Using these three components, we have modelled 'Severe Weight Loss' as a post-coordinated concept defined by: a subclass of "*Weight Loss Finding*" that has "*Severe*" as the range of its object property: '*Severity*'. Another similar example is "*Wedge Resection of Lesion of Lung*", which is a surgery type in LUCADA. We modelled this in SNOMED-CT as a "*Wedge Resection*" '*ProcedureSite*' "*Lesion of Lung*", where "*Wedge Resection*" and "*Lesion of Lung*" are classes and '*ProcedureSite*' is an object property used to specify the body structure affected by a procedure.

As the reader may readily appreciate, this ability to construct post-coordinated and complex concepts by using the SNOMED-CT classes and relationships as our building blocks greatly enhances the ontology designer's ability to semantically model complex concepts while staying within the boundaries of our information standard. Having said this, some data items were just too complex to be represented within the vocabulary of SNOMED-CT. As a result, we had to add 13 new classes to the LUCADA ontology.

An example of such a class is "*Induction Chemotherapy to downstage before surgery*", which appears as a chemotherapy type in the LUCADA database. Since modelling this as a post-coordinated concept was not feasible, a concept with the same name was added to the LUCADA ontology. It is worthwhile to mention that for this very concept, a concept addition request was also made to the SNOMED-CT UK Terminology Centre. After their review, the request was accepted on July 8 2011 and the concept was added in the October 2011 SNOMED-CT release.

The resulting classes and properties constituted the clinical domain of the LUCADA ontology, enabling a semantically accurate mapping with all entities in the LUCADA concepts list. Figure 3.7 depicts the ontology's T-Box, including only the most essential

classes and object and datatype properties to preserve visual clarity. At this stage, the ontology consisted of 376 classes, 37 object properties and 63 datatype properties.

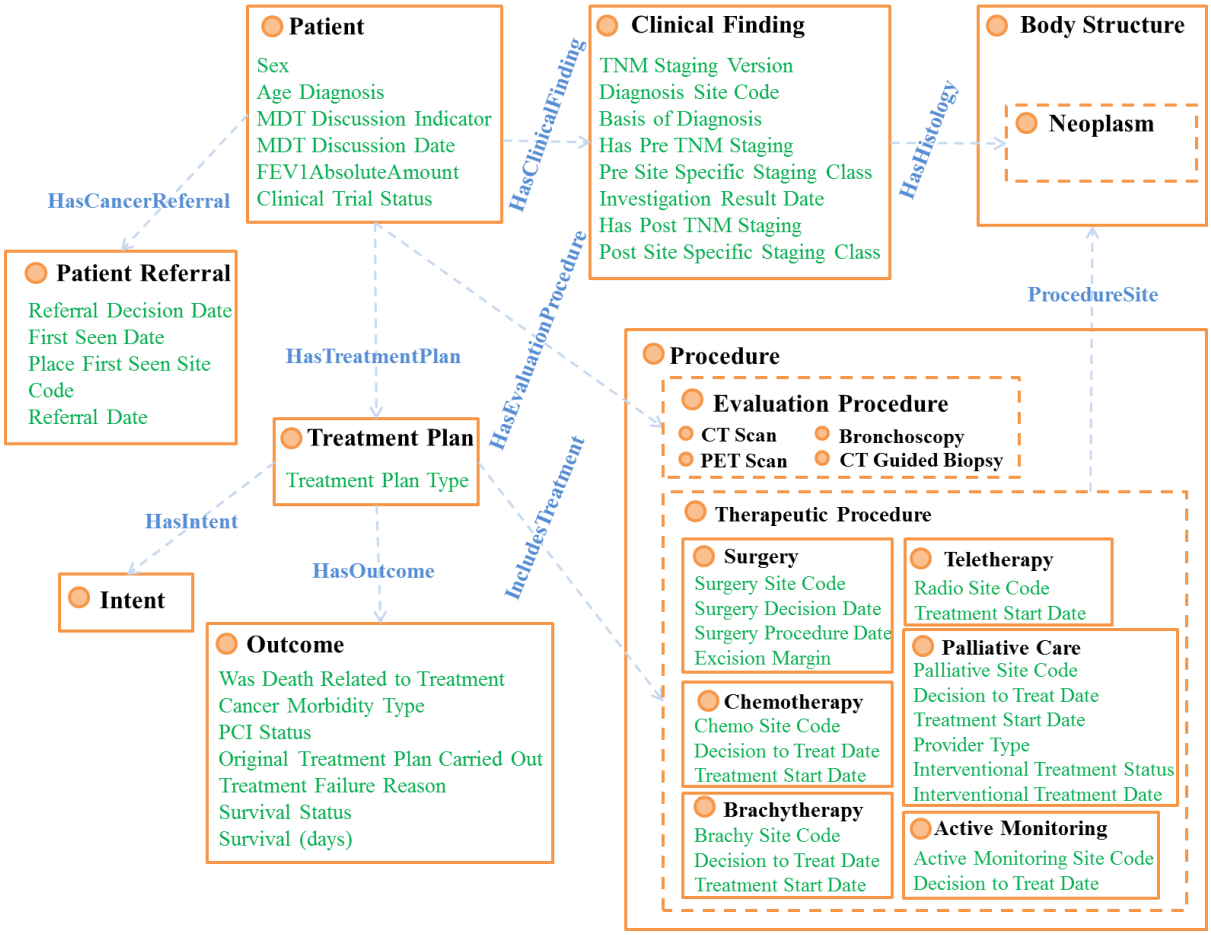

Figure 3.7: The LUCADA ontology clinical domain T-Box. The orange circles represent SNOMED-CT classes, the blue edges represent object properties between these classes and the green list items represent the datatype properties belonging to the respective classes.

A side by side comparison of the T-Box depicted in Figure 3.7 and the LUCADA data model in the previous section reveals that we chose to represent LUCADA concepts either as classes or (object or datatype) properties between classes in SNOMED-CT. In other words, the classes in Figure 3.7 were added either to directly represent LUCADA entities or as suitable SNOMED-CT classes which could accommodate LUCADA entities as their object or datatype properties.

This mapping of entities into ontology properties can best be explained by way of examples. For instance, the LUCADA ontology models the LUCADA data item ‘Sex’, thorough an identically named datatype property, added to the Patient class (Figure 3.7).

Another way of modelling patient gender could have been by defining an object property '*hasGender*' between the "*Patient*" and "*Patient Sex*" classes in SNOMED CT. However, this would have required the addition of another class to the LUCADA ontology and was not preferred due to the fact that 'Sex' was regarded as a leaf entity, that is, there were no other entities that could be modelled through distinguishing "*Patient Sex*" as a separate class. Similarly, most data items with Boolean (True / False) values, e.g. 'Staging Procedure Performed?', 'Was Death Related to Treatment?', were considered as leaf entities and represented as datatype properties with Boolean ranges. Object properties were adopted only in the following three situations:

1) When the creation of a distinctive class for an entity could be beneficial to model other entities through the newly created class properties. An example of this is the '*hasTreatmentPlan'* object property, which has domain "*Patient*" and range "*Treatment Plan*". Here, the addition of "*Treatment Plan*" as a separate class enables connecting it to individual treatment types under Procedure with the '*includesTreatment*' object property. This structure enables modelling a patient with a suggested treatment plan which includes various distinct treatment types.

2) When more than one data item could be grouped under a common parent concept. An example of this is the introduction of the SNOMED-CT concept "*Clinical Finding*", which subsumes all diseases in LUCADA such as 'Primary (Cancer) Diagnosis', 'Dementia', 'Cardiovascular Disease' and other comorbidities. This allows use of a single object property, '*hasClinicalFinding*', to connect a "*Patient*" individual to all (taxonomically) disease-related concepts in LUCADA.

3) When suitable object properties already existed within SNOMED-CT. While building the LUCADA ontology, precedence was given to using the properties inherently defined in SNOMED-CT. These are originally called defining attributes but translate into object properties in the OWL-2 representation of SNOMED-CT. Above, we have already given

two examples of how we made use of these object properties to construct post-coordinated concepts such as "*Severe Weight Loss*". Currently, there are over 50 defining attributes to model concept relations within SNOMED-CT [103]. Two examples are: '*HasIntent'*, used to indicate the "*Intent*" of a "*Procedure*", and '*ProcedureSite'*, used to specify the "*Body Structure*" affected by a "*Procedure*".

In terms of ontology design heuristics, we adopted a "bottom up" approach, which involves starting by modelling the most specific concepts and gradually grouping them into super classes by taxonomically traversing upwards. Since we based our ontology on SNOMED-CT and adopted a bottom up design approach, whenever we added a new class to the LUCADA ontology, we incorporated its SNOMED-CT taxonomy (to allow semantic similarity) and class names (to improve lexical similarity) into our ontology. In practice, this meant that whenever we added a new class, we also added its parent classes into the LUCADA ontology. As an example, in order to add the "Neoplasm" class that subsumes tumours of different histology types, we added all parent classes of "Neoplasm" from the SNOMED-CT taxonomy into the LUCADA ontology, where all class names are copied verbatim from SNOMED-CT. This is depicted in Figure 3.8.

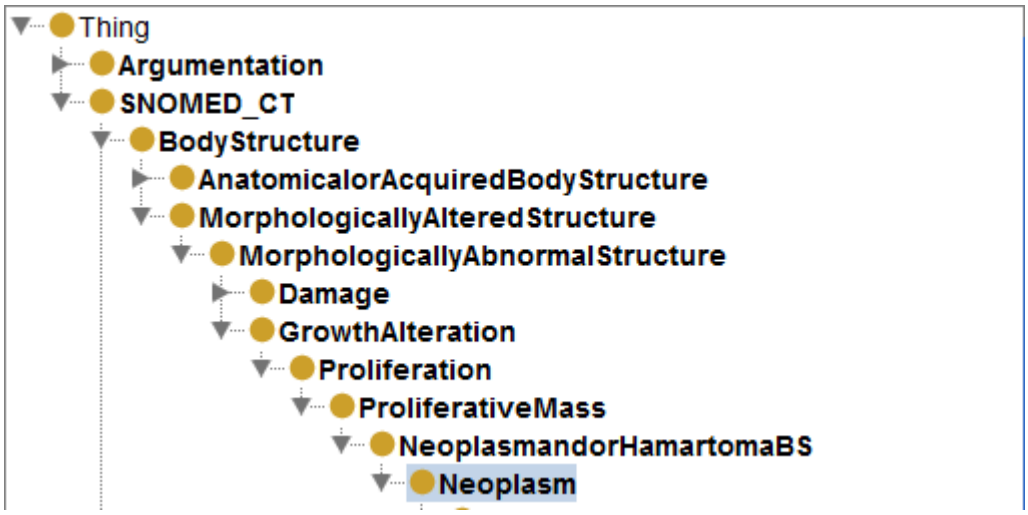

Figure 3.8: The super classes of "Neoplasm" in the LUCADA ontology, as directly copied from the SNOMED-CT taxonomy.

Overall, during the design of the LUCADA ontology, we followed the main design heuristics set out in [169]. The LUCADA ontology is designed as a domain-specific and compact knowledge base in OWL-2. It strictly follows the taxonomical structure of SNOMED-CT. In terms of its internal concept structure, which represents non-taxonomical relationships between different concepts (spatial, causal, functional and so on); it endeavours to make use of native SNOMED-CT relationships as much as possible. However, in cases where the SNOMED-CT relationships cannot express the intended semantic relations (as showcased in Figure 3.7), proprietary object and data properties are added.

3.3.4. LUCADA Ontology Module Extraction

Having built the LUCADA ontology manually, we re-evaluated the feasibility of using an ontology matching tool to find accurate mappings between the LUCADA ontology and SNOMED-CT and output a domain-specific SNOMED-CT module based on these mappings. As discussed in the previous section, during its design we paid special attention to the taxonomy and nomenclature of the LUCADA ontology to follow SNOMED-CT as closely as possible. This was motivated by our goal to retrospectively map the LUCADA concepts to SNOMED-CT and extract a domain specific module as smoothly as possible.

In order to retrospectively map the LUCADA ontology to SNOMED-CT, and similar to the experiments described in section, 3.3.2.2, we used LogMap-2 with and without user interaction. While our target ontology, i.e. the OWL-2 version of SNOMED-CT, remained the same, for this set of experiments we used the manually designed LUCADA ontology as our input. The results of these experiments are as given in Table 3.7. These results have been published by Jimenez-Ruiz et al [179].

	# LUCADA Classes	**# Output Mappings**	**# Correct Mappings**
Without User Interaction	376	305	260

With user Interaction	376	279	259

Table 3.7: The results of ontology matching between the manually designed LUCADA ontology and the SNOMED-CT.

As can be seen from this table, the recall rates both with (0.74) and without (0.81) user interaction were significantly higher compared to the initial results. This was encouraging since it indicated that our efforts in adopting the SNOMED-CT nomenclature and taxonomy while designing the LUCADA ontology paid dividends. In both cases, the concepts that could not be directly mapped to SNOMED-CT were those which either did not have a one-to-one mapping (post coordinated classes) or did not exist in SNOMED-CT (proprietary classes) as explained in section 3.3.3.

However, an equally –if not more- important performance metric was the mapping accuracy since without a satisfactory level of accuracy, a high recall rate did not mean much. For this purpose, we manually assessed the recommended mappings for both experiments and found that the mapping accuracy was approximately 0.85 (260/305) without user interaction and around 0.93 (259/279) with user interaction. Similar to our initial results, user interaction during mapping improved the reliability of the suggested mappings by LogMap-2.

The next step was to extract a minimal and complete module of SNOMED-CT that preserved all semantic information relevant to our mapped concepts. For this purpose, we input the mappings acquired with user interaction into the Locality Module Extractor tool [186], which is the software implementation of the DL-logic-based module extraction notions described in [166].

The output of the Locality Module Extractor was a SNOMED-CT module with a taxonomy that, unsurprisingly, looked very much like that of our LUCADA ontology. However, as a ‘safe’ module, this ontology had a wider scope with 1160 classes, and 731 equivalent class axioms, which were directly retrieved from SNOMED-CT. A

demonstrative Protégé view of the module taxonomy (on the left) and the class equivalence axioms (on the right) for the "Neoplasm by body site" class are given in Figure 3.9.

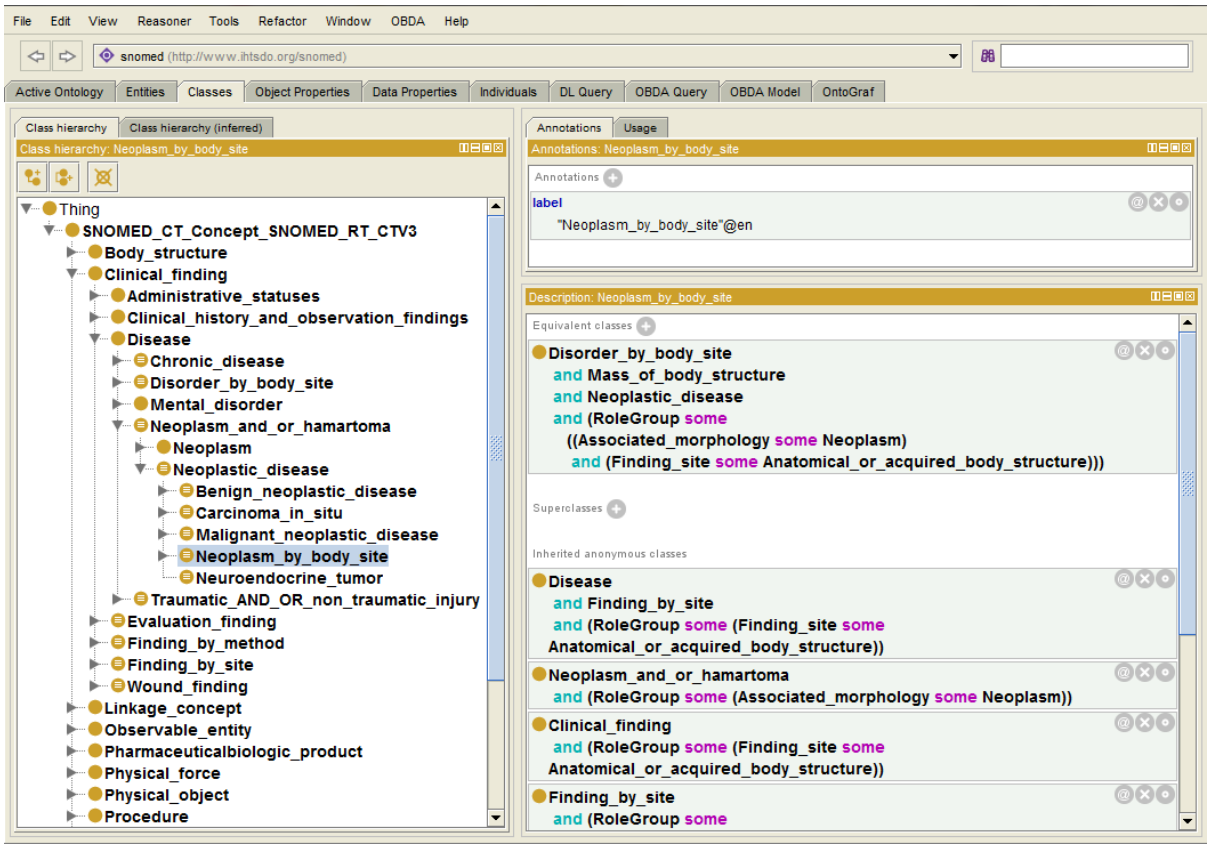

Figure 3.9: The class hierarchy view on the left shows the taxonomy of the extracted SNOMED-CT module. On the right, the equivalence class axioms of the selected "Neoplasm by body site" class are listed.

There are two major advantages to extracting and making use of this module. First, it ensures that our knowledge base adheres to SNOMED-CT. Second, the capability to reuse the rich semantic clinical information that is already stored within SNOMED-CT minimises our design effort and the margin for errors. As discussed previously, while part of this 'clinical information' is the domain taxonomy, a significant part is actually the concept structure, i.e. the non-taxonomical relations between concepts, such as the class equivalence axioms of the "Neoplasm by body site", as showcased in Figure 3.9.

While the extracted module was satisfactory, as the recall rates of our ontology matching experiment indicate, this module did not account for all domain concepts and properties within the LUCADA ontology. Therefore, instead of making use of this module on its own, we needed to import it and its respective mappings from the ontology mapping step

into the LUCADA ontology to acquire an integrated ontology that involves all LUCADA entities along with their SNOMED-CT mappings. The overall process can be better summarised with a process diagram which depicts the individual operations with their inputs and outputs as given in Figure 3.10 below.

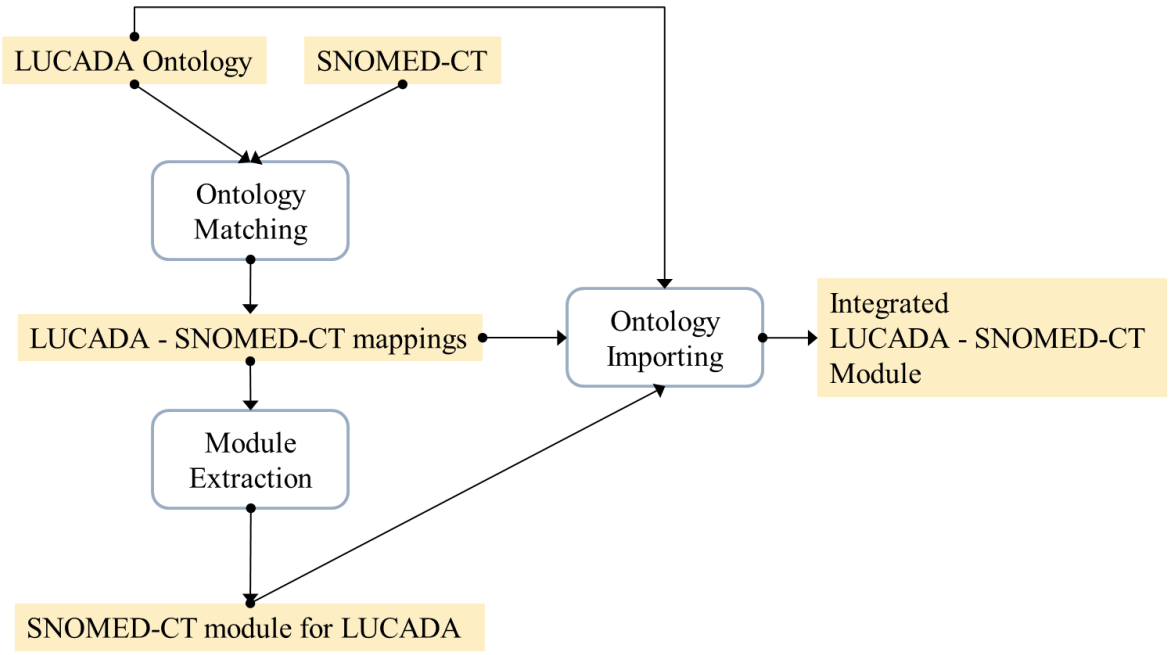

Figure 3.10: The process diagram that summarises the major ontological operations in attaining the Integrated LUCADA-SNOMED-CT module. The ontology matching operation has been carried out with LogMAp-2 and module extraction has been done with Locality Module Extractor software.

The resulting Integrated LUCADA-SNOMED-CT ontology consists of 1579 classes, 63 object properties, 63 data properties and 1041 equivalent class axioms, part of which comes directly from SNOMED-CT (such as the class equivalence axiom for "Neoplasm by body site" class in Figure 3.9) and part of which exist to formally represent the concept mappings that are uncovered during the ontology matching step.

Overall, this integrated ontology has major benefits in terms of enhancing interoperability, minimising design efforts and inheriting a semantically much richer knowledge base. However, it should also be noted that these advantages come at a price. The integrated LUCADA-SNOMED-CT ontology is significantly larger in size compared to our manually designed LUCADA ontology. Table 3.8 below gives a side by side comparison of the size

of the manually designed LUCADA ontology and the integrated LUCADA-SNOMED-CT ontology.

	Classes	Object Properties	Data Properties	Subclass Axioms	Equivalence Axioms
LUCADA ontology	376	37	63	386	0
SNOMED-CT LUCADA module	1160	26	0	595	731
Integrated LUCADA – SNOMED-CT Ontology	1579	63	63	999	1041

Table 3.8: Comparison of the manually designed LUCADA ontology with the SNOMED-CT LUCADA module and the Integrated LUCADA - SNOMED-CT ontology.

3.4. Discussion

In their 2007 paper, Hu et al. [187] discuss the major differences between reference and domain-specific ontologies. They point out that while the developers of application-independent reference ontologies (such as SNOMED-CT) can attach a high level of philosophical sophistication to their approach, those like us -who need to build more niche ontologies- are usually faced with additional concerns to meet application and domain-specific requirements and eventually modelling the 'reality' of the given domain.

In our ontology design effort, we attempted to strike a balance between generality and specificity and avoid the very common (to the extent of being the norm!) pitfall of designing our own purpose-built, ad-hoc ontology for reasons of convenience and controllability [188]. Deciding what concepts to exclude in a knowledge-base that represents a rich and multidisciplinary domain as lung cancer treatment is a challenging task on its own. In this respect, our domain elicitation effort was greatly facilitated by the availability of the LUCADA database model on which we based our conceptualisation.

Many researchers agree that designing a well-developed and practically usable ontology is a challenging task [169]. While there are well established heuristics to guide the developer, ontology design is a highly creative and iterative process and in a way more art

than science. Between 17 February 2011 and the submission of this thesis, the clinical domain of the LUCADA ontology has gone through more than 60 iterations, before taking its form at the time of submission.

An interesting research issue in ontology engineering is the formal evaluation of an ontology [189]. Since ontologies can be built for different purposes and with different design concerns, the evaluation criteria vary widely with respect to the use case. While there is no established set of guidelines for evaluating ontologies, the iterative improvements are mainly motivated by maximising domain coverage and practical usefulness.

In this chapter, we covered all design stages of the LUCADA ontology, from domain conceptualisation to yielding accurate mappings with SNOMED-CT for the purpose of extracting an efficient and succinct SNOMED-CT module and integrating it with our manually designed ontology. In terms of domain coverage, the LUCADA ontology met the design goal of expressing the clinical LUCADA data model as closely as possible. In the next chapter, we will talk about how the ontology is extended to incorporate guideline rule knowledge and focus more closely on practical implementation issues such as ontological inference performance, expressivity and scalability.

Chapter 4 - Guideline Rule Inference Framework

In the previous chapter, we discussed the design process we followed in creating an integrated LUCADA - SNOMED-CT ontology based on the flat and tabular LUCADA data model. The resulting ontology consisted entirely of clinical classes and properties inherited from, and mapped to, SNOMED-CT wherever possible. In this chapter, we expand the LUCADA ontology in order to incorporate guideline rule knowledge and introduce a framework for determining the guideline eligibility of patient records through ontological inference. Following this, we evaluate and compare the scalability and inference performances of two strategies for the implementation of the guideline rule inference framework.

Clinical ontologies can be utilised to form the knowledge basis of information-intensive applications [187]. They are particularly well-suited to classify and encode semantic relations between concepts within their domains. Earlier in Chapter 2, we introduced the concept of OWL "defined classes" that have necessary and sufficient class equivalence axioms and showcased how the ontological inference operations, namely 'classification' and 'realisation', can be performed to uncover implicit class membership (individual to class) and class subsumption (subclass to superclass) relations in an ontology. An example of realisation was given in Figure 2.5 in Section 2.2.4.

Based on the discussions in Chapters 2 and 3, we chose to model our guideline rule knowledge within our integrated LUCADA – SNOMED-CT ontology in OWL-2. However, before demonstrating how an OWL-2 ontology can be utilised to capture and interpret guideline rule knowledge, it is worthwhile to analyse the anatomy of a representative guideline rule and briefly discuss how various CIG formalisms encode this knowledge.

4.1. Background

As a typical example, we analyse the lung cancer care guideline rule below, which is taken from the National Institute for Clinical Excellence (NICE) Lung Cancer 2011 Guideline document [3]:

> *"Offer chemotherapy to patients with stage III or IV NSCLC and good performance status (WHO 0, 1 or a Karnofsky score of 80–100)."* **(Rule 1)**

In rule-based deduction systems, the convention is to refer to each IF pattern as an antecedent (body) and to each THEN pattern as a consequent (head). Hence, we can formally break 'Rule 1' into two functional components: 1) the antecedent, which specifies the rule eligibility criteria; and 2) the consequent, which specifies the action(s) to take when the conditions in the antecedent are satisfied [190]. In this particular example, the guideline rule consequent entails "offering chemotherapy" to a set of patients that satisfy specific criteria encapsulated within the rule antecedent.

All CIG formalisms have different modelling constructs in order to capture such criteria to trigger the execution of an encoded guideline. While the names vary from formalism to formalism, they all serve the same elementary purpose to specify an abstract patient state that a particular guideline rule addresses. In GLIF 3, these are called *patient state steps* that serve as entry points into a guideline; in EON they are called *scenarios*, enabling a patient's automatic entry into the appropriate guideline plan; in SAGE, an approximate equivalent of these are called *context nodes* which trigger guideline execution.

Due to their suitability for the same task, OWL and OWL-2 have been utilised to encapsulate rule criteria and other context-aware specifications in various research previously. As a matter of fact, Strang et al. [191] report that ontologies are the most expressive models for designing context-aware systems.

In 2006, Kashyap et al. [192] proposed simplifying a rule-base by making use of defined OWL classes to capture guideline eligibility criteria. According to their framework, the hardcoded rule-base could reference OWL classes that are defined and maintained as part of an ontology. While their solution is elegant, it relied on a separate rule base which needed to be maintained in tandem with the ontology. As a consequence of this, they reported that "*the introduction of OWL definitions has a negative impact on rules execution performance due to the round trip costs between the commercial rules engine and the ontology*". Furthermore, since datatype range restrictions, e.g. Age < 40, were not allowed in OWL, the authors needed to work around this issue by making use of a commercial OWL engine named Cerebra to introduce custom data types mapped to their XML schema.

In 2008, Austin [18] proposed another ontology-based solution to the problem of representing guideline rule criteria. In order to power a clinical decision support application for colorectal MDT's, they built a proprietary OWL ontology, which represents the domain of colorectal cancer, and made use of separately stored SPARQL [193] queries to specify patient cohorts that fulfil a guideline criteria. As mentioned in Chapter 2, OWL and OWL-2 are based on the RDF schema and SPARQL is the W3C endorsed query language for RDF graphs. Hence, SPARQL queries are limited to make use of only the RDF triplet structures in OWL, and cannot interpret the Description Logic axioms. Consequently, Austin's solution approach necessitated the reduction of an OWL ontology to an RDF triplet table, which is structurally very similar to a relational database table that can be queried in SQL. The RDF table was then queried by separately stored SPARQL queries which represented guideline rule criteria. The only advantage of this approach over an SQL query was that the subclass-superclass relationships, e.g. 'Colorectal cancer is a Cancer', were retained in the RDF triplet table and could be used in the queries.

In his doctoral thesis [194], Williams incorporated a domain specific proprietary breast cancer OWL ontology with an extension to the defeasible extended logic programming (DELP) framework. In this work, Williams made use of clinical trial results to produce rules and harnessed the ontology to encode rule eligibility criteria, i.e. rule antecedents, similar to Kashyap's work described above. Once the rule eligibility was inferred through ontological inference, the resulting rules were used to create arguments similar to the argumentation framework of PROforma [32]. While the major contribution of this work was in extending an existing argumentation framework that was kept separate from the ontology, it is still relevant to our guideline rule inference framework since it utilised DL inference in an OWL ontology to answer A-Box conjunctive queries, which indicated eligibility for a clinical trial rule.

In a more recent study [195], Beimel and Peleg proposed using OWL-based inference and SWRL rules to manage software data access policies in healthcare organisations. While this work is not directly related to clinical guidelines, their approach mimics decision rules and can be applied to represent guideline eligibility criteria instead of data access policies. The central concept underlying their model is 'Situation', which specifies an end-user's data-access scenario. 'Situations' are encoded as defined OWL classes that capture specific access scenarios in the form of class equivalence axioms in the ontology. According to their model, each incoming data-access request is formalised as an OWL individual with appropriate property values to represent the characteristics of the access request. A semantic reasoner is run to perform 'realisation' on the ontology. If the OWL individual is inferred to be a member of one of the Situation defined classes, then its response is accepted to be 'permission granted' for that particular scenario. While the underlying approach is powerful, this was a proof of concept study where the access requesting OWL individuals were manually included in the ontology and the practical aspects of the software implementation stages, such as scalability, were out of scope. In addition, the

utilisation of the SWRL engine was required as an interim step to infer ‘R2R relation’ individuals without which the data-access request individual could not be realised as a member of a Situation class.

In summary, the originality of these approaches was formalising decision or rule criteria in the form of defined OWL classes in an ontology. According to this, upon addition of OWL individuals to the ontology, a semantic reasoner is run to realise the individuals and infer which decision/rule criteria a particular individual satisfies. A common aspect of these methodologies is that they all rely on a separately maintained rule base and rule engine for formalising the actions to take when eligibility criteria are satisfied. Consequently, the addition of a separate rule engine adds an interim step to the inference process and affects the performance negatively. Furthermore, to our knowledge, none of the previous research which adopted this realisation-based inference framework focused on the implementation and scalability issues.

Motik et al. report that a combination of OWL-DL and rules may be desirable for applications in the Semantic Web. However, it may also easily lead to undecidability of interesting reasoning problems unless the rules are ‘DL-Safe’ [196]. Our prior experience with a proof of concept prototype that combines SWRL rules and Description Logics also revealed that the addition of rules into a DL framework complicates the inference process and affects system performance negatively [68].

In the guideline rule inference framework described here, we adopt a similar approach to previous research for encoding rule eligibility criteria as defined OWL classes. However, due to the performance issues involved, we choose not to utilise an external rule base and engine for encoding what actions to take upon rule eligibility. Instead, we opt to store the rule consequents within the ontology as well. This results in a purely ontological approach, which necessitates expanding the LUCADA ontology with additional class and properties that are necessary to represent guideline rules ontologically.

4.2. LUCADA Ontology Argumentation Domain

The integrated LUCADA – SNOMED-CT ontology, introduced in the previous chapter, contained only clinical classes and properties, which were inherited from (and mapped to) SNOMED-CT wherever possible. In this section, we discuss how the LUCADA ontology has been extended to include an 'Argumentation' subdomain to act as the basis of an extensible guideline rule inference framework for our intended clinical decision support tool: Lung Cancer Assistant.

In order to keep the SNOMED-CT classes that represent the clinical domain separate, these new classes were added under a separate subdomain class named 'Argumentation'. In its current form, the argumentation classes consist of 'Decision' and the hybrid class "Patient Scenario", which is subsumed by both the 'SNOMED-CT' and 'Argumentation' subdomain classes. The "Patient Scenario" class is at the heart of the guideline rule inference framework, encapsulating the eligibility criteria for a guideline rule in the form of a class equivalence axiom. Conceptually, it can be considered as a defined "Patient" class, which describes an abstract patient group that fulfils a guideline rule's eligibility criteria. Compared to CIG formalisms, the "Patient Scenario" class represents the analogue of a patient state step in GLIF3, a scenario in EON and a context node in SAGE.

The major classes and properties, which collectively constitute the T-Box, in this extended ontology are depicted in Figure 4.1. In this figure, the newly added argumentation classes are colour-coded in black and the clinical SNOMED-CT classes are colour-coded in orange.

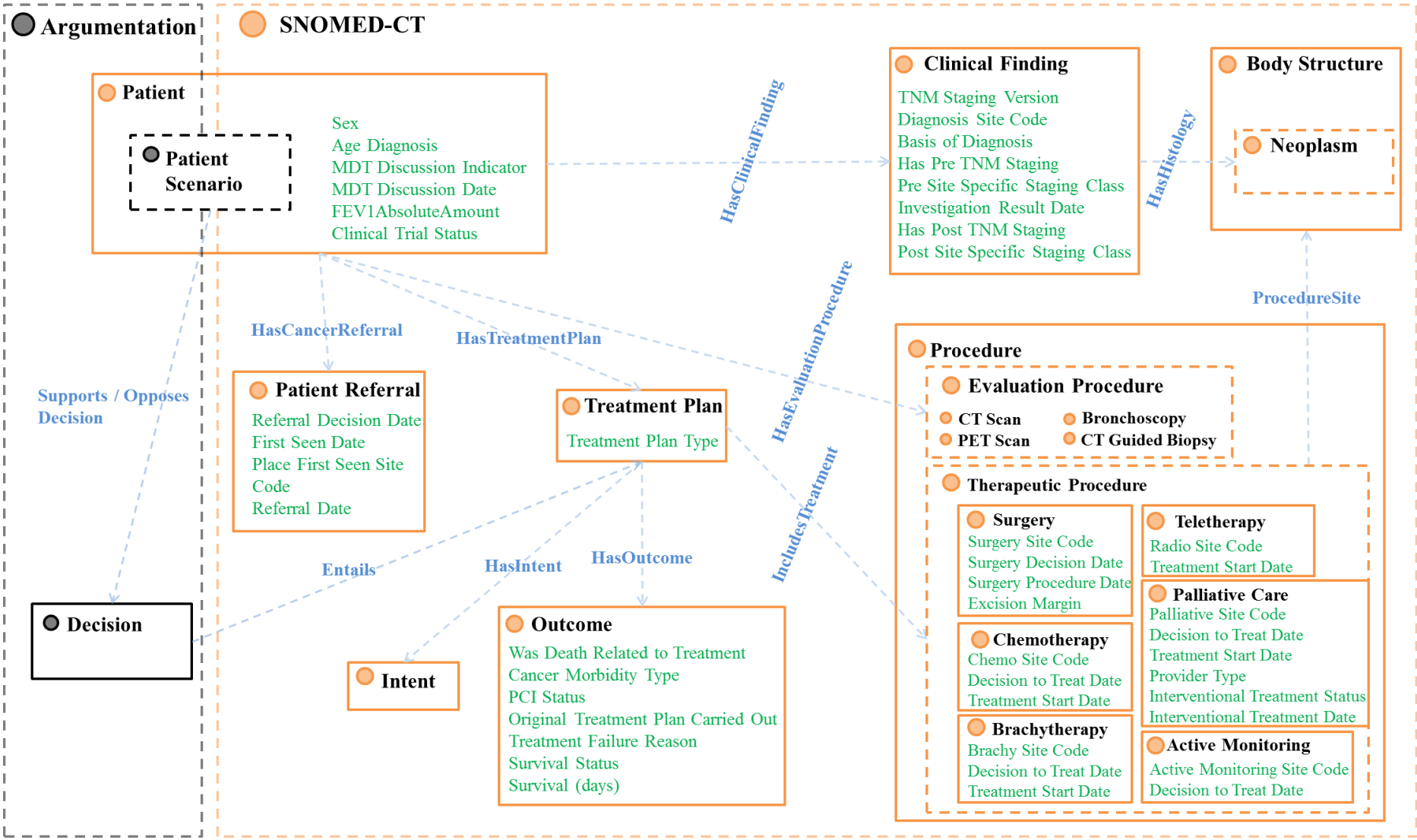

Figure 4.1: The primary classes and properties in the LUCADA ontology T-Box. The orange circles denote 'SNOMED-CT' classes and the black circles denote 'Argumentation' classes. The blue edges stand for object properties between any two classes and the green list items represent the datatype properties belonging to their respective class.

The DL expressivity of this resulting OWL-2 ontology is ALCHIQ(D), which is a functional subset of the more general SHIQ(D) expressivity of OWL-2 that disallows transitivity axioms [196]. We can best demonstrate how the Argumentation subdomain classes and properties are utilised for guideline rule inference by using the rule given in the beginning as an example.

To include this guideline rule in our ontology, we need to add a new (defined) "Patient Scenario" class, whose equivalence class axiom encapsulates the rule eligibility criteria, i.e. antecedent. We refer to this class as "Patient Scenario 1" and encode its class equivalence axiom as:

> "Patient and (hasPerformanceStatus some (WHOPerformanceStatusGradeO or WHOPerformanceStatusGradeI)) and (hasClinicalFinding some (NeoplasticDisease and ((hasPreTNMStaging value "IIIA") or (hasPreTNMStaging value "IIIB") or (hasPreTNMStaging value "IV")) and (hasPreHistology some NonsmallCellCarcinoma)))" **(Axiom 1)**

In this example, (Axiom 1) is a compound class equivalence axiom that combines various atomic expressions through conjunction (and) and disjunction (or) operators, colour-coded in turquoise. The words in pink represent existential (some) and value (value) restrictions. Axiom 1 can be translated into plain English as: '*Patients whose performance status are either 0 or 1 and who have Neoplastic Disease with TNM staging of either 3 or 4 and histology type non-small cell carcinoma'*.

As showcased earlier by the frameworks proposed by Kashyap et al. [192], Beimel et al. [195] and [197], when a new OWL individual is added to the ontology, a semantic reasoner can 'realise' this individual to determine whether or not it is a member of this particular "Patient Scenario" class. This inferred class membership of an OWL individual to a specific "Patient Scenario" implies that the individual, which represents a patient record, satisfies the eligibility criteria of a "Patient Scenario" and is therefore subject to the recommendations or actions specified by the corresponding guideline rule.

Within our framework, the rule-specific recommendations, i.e. the consequents, are stored internally in the form of chained binary relations within the ontology, whereby each "Patient Scenario" class is related to a decision class through a 'supportsDecision' or 'opposesDecision' property and each "Decision" 'entails' a specific "Treatment Plan". To give a concrete example, the consequent of Rule 1 is modelled as depicted in Figure 4.2, where "Patient Scenario 1" supports "Chemotherapy Plan Decision", which in turn 'entails' that a "Chemotherapy Treatment Plan" be given to the eligible patient.

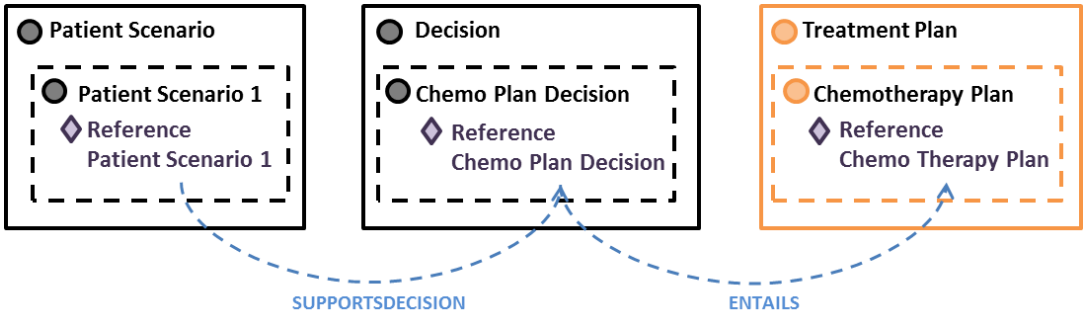

Figure 4.2: A reference-individual-level connection between a Patient Scenario, a Decision and a Treatment Plan, which represents the rule consequent statement "Offer Chemotherapy". In the figure, OWL classes are denoted as circles and OWL individuals are denoted as diamonds.

This relation is explicated within the A-Box of the ontology, using the "Reference individuals" of the classes involved. For our purposes, a "Reference Individual" is an ontological modelling feature and is used to model the unique instantiation of an abstract and intangible class within the ontology. For instance in the LUCADA ontology, "Patient Scenario", "Decision" and "Treatment Plan" are all abstract classes and therefore contain reference individuals that are used for representing their standard values.

4.3. Guideline Rule Inference Framework

In more formal terms, a guideline rule R(i) can be broken down into its antecedent Ant(i) and consequent Con(i) parts. According to the guideline rule inference framework, within a given ontology Ont, Ant(i) is formalised as a patient scenario class PS(i), and Con(i) is captured with the help of chained binary relations between the reference individuals of the corresponding "Patient Scenario", "Decision" and "Treatment Plan" classes as depicted in

Figure 4.2. The overall guideline rule knowledge is stored in the ontology T-Box, such that ∀ R(i) ∃ {PS(i) ⇒ Con(i)} ∈ Ont. As a result, for N guideline rules, the ontology consists of a set of patient scenarios classes $\widehat{PS}$ = {PS(1), PS(2),…, PS(N)} and a set of chained binary relations that represent the rule consequents $\widehat{Con}$ = {Con(1), Con(2), …, Con(N)}.

At this stage, for brevity, we establish some notations for the classification and realisation tasks carried out by the semantic reasoner. Consider the ontology 'Ont' that includes two separate Patient Scenario classes PS(i) and PS(j) and a Patient individual Ind(k). When a semantic reasoner is run, if PS(i) is classified under PS(j), we will denote it as "PS(i) ☪ PS(j)". Also, if Ind(k) is inferred to be an individual of (realised as) PS(i), it will be denoted as "Ind(k) $\overset{\hat{I}}{\rightarrow}$ PS(i)".

After the guideline rule knowledge has been added as {$\widehat{PS}$, $\widehat{Con}$} ∈ Ont, we can add a new Patient individual Ind(k) to the ontology and run the semantic reasoner to evaluate Ind(k) against $\widehat{PS}$ and determine the subset $\widehat{PS}$(k)={ $\widehat{PS}$: Ind(k) $\overset{\hat{I}}{\rightarrow}$ *PS(k)* }, which contains all patient scenarios that Ind(k) is realised as. Once $\widehat{PS}$(k), i.e. the set of patient scenarios that a given individual is eligible for, is inferred, we can retrieve the corresponding Con(k) for each PS(k) ∈ $\widehat{PS}$(k), to obtain $\widehat{Con}(k)$={ $\widehat{Con}$: PS(k) ⇒ Con(k), PS(k) ∈ $\widehat{PS}$(k)}, which contains the recommended actions of all guideline rules that are applicable to the patient individual Ind(k). In summary, given $\widehat{PS}$, $\widehat{Con}$, Ind(k) ∈ Ont, the guideline rule inference described here proposes a methodology to uncover the implicit knowledge $\widehat{Con}(k)$.

4.3.1. Ontological Characteristics of the Framework

The ontological knowledge captured through this framework has some key characteristics which are listed below:

Characteristic 1: The guideline knowledge is incomplete.

This feature highlights that we cannot guarantee for each given patient individual, Ind(i), to be realised as at least one Patient Scenario. Or more formally, $\forall$ Ind(i) $\nexists$ PS(i) $\in$ Ont, such that Ind(i) $\overset{\hat{I}}{\rightarrow}$ PS(i). This implies that our ontological knowledge is incomplete and that some patient individuals may not fall under any patient scenario class in the ontology. The incompleteness may be due to the absence of a relevant guideline rule in the ontology or due to the 'Patient' individual missing some key properties, such as tumour staging or histology type. In the former case, the ontology lacks the asserted class axioms and in the latter the individual lacks the vital property constraint axioms to infer new knowledge. This is one of the major shortcomings of rule-based decision support and will be discussed in more detail in Chapter 5.

Characteristic 2: The guideline knowledge is neither minimal nor exclusive.

According to this characteristic, it is possible for the eligibility criteria, i.e. the equivalent class expression, of a patient scenario to be logically subsumed by the eligibility criteria of a different patient scenario. More formally, for two 'Patient Scenario' classes PS(i) and PS(j), such that PS(i) $\neq$ PS(j), both PS(i) $\subset$ PS(j) and PS(j) $\subset$ PS(i) are possible. This implies that the guideline rule eligibility knowledge stored in 'Patient Scenario' classes does not have to be minimal or exclusive. Ontologically, this characteristic is undesirable since it allows replications in the knowledge base. However, it is necessary to account for such replications that naturally occur between guideline rules even within a single guideline document.

Characteristic 3: The guideline knowledge can contain conflicting information.

This characteristic indicates that different guideline rules, for which a patient is eligible, may have conflicting 'consequents'. Remembering our definition of a guideline rule as R(a) = {Ant(a), Con(a)}, and the set of binary relations (Figure 4.2) that we utilise to store the rule consequents, we can postulate that $\forall$ Con(i) $\exists$ Arg(i), such that Con(i) $\vDash$ Arg(i). In

this setting, the third characteristic entails that given a set of applying rule consequents $\widehat{Con}(i)$ = {Con(a), Con(b)} for a 'Patient' individual Ind(i), it is possible for Arg(a) and Arg(b) to have conflicting claims. This is due to the often incomplete and inconsistent nature of medical knowledge, which we handle by adopting a similar approach to the argument-based decision model of [32] introduced in Chapter 2. According to this, we evaluate our guideline rule knowledge against a patient record to automatically construct patient-specific arguments that support or oppose available treatment plan classes, i.e. decision options.

In case R(a) and R(b) address precisely the same patient cohort, i.e. PS(a) = PS(b), their conflict represents contradictory clinical knowledge. Such a situation is not uncommon, especially if R(a) and R(b) originate from different guideline documents. On the other hand, if PS(a) ≠ PS(b), the conflicting Arg(a) and Arg(b) do not necessarily imply an inconsistency. In either case, automatic conflict resolution is not included within the scope of the guideline rule inference framework since our aim is not to build an autonomous system that performs all inference internally and makes the decision on behalf of the clinicians. Previous research also leads to the conclusion that the complexity and uncertainty inherent in clinical knowledge makes it highly unlikely that such completely automated decision systems will succeed [111]. Therefore, we opt to preserve and display all recommendations/arguments – be they conflicting or reinforcing- in order to objectively inform the clinicians' decision making.

In addition to these three knowledge representation characteristics, our framework differs from those discussed in the previous section in that we store and process the rule consequents, i.e. $\widehat{Con}$, as part of the domain knowledge within the ontology, instead of using an external rule-base and rule engine for this purpose. This design choice was made in order to eliminate the need to simultaneously maintain a rule-base and so to introduce an

interim rule inference step, which would complicate the process and slow the inference performance down.

4.3.2. Patient Similarity Measure

Another convenient feature of utilising an ontological representation for inferring guideline rule eligibility is that 'Patient' individuals (or class descriptions), which satisfy semantically identical criteria, are automatically inferred to be realised (or classified) under the same 'Patient Scenario' classes. This allows measuring the relative similarities of different individuals to a single reference individual. According to this, an increase in the number of common parents between the reference individual and another individual implies an increase in the number of common semantic constraints that they jointly satisfy.

Figure 4.3 illustrates a demonstrative scenario for a reference individual Ind(A), where only a closed set of 4 'Patient Scenarios' have been modelled in the ontology. In the figure, the purple diamonds represent 'Patient' individuals and the numbers within the diamonds indicate the number of common 'Patient Scenario' parents that individual shares with Ind(A). Focusing on Ind(A) , Ind(B) and Ind(C), we can formulate the inferred ontology as below:

Ind(A) $\overset{\hat{I}}{\rightarrow}$ $\widehat{PS}$(A), where $\widehat{PS}$(A) = {PS(1), PS(2), PS(3), PS(4)}

Ind(B) $\overset{\hat{I}}{\rightarrow}$ $\widehat{PS}$(B), where $\widehat{PS}$(B) = {PS(1), PS(2), PS(3), PS(4)}

Ind(C) $\overset{\hat{I}}{\rightarrow}$ $\widehat{PS}$(C), where $\widehat{PS}$(C) = {PS(1), PS(2), PS(4)}

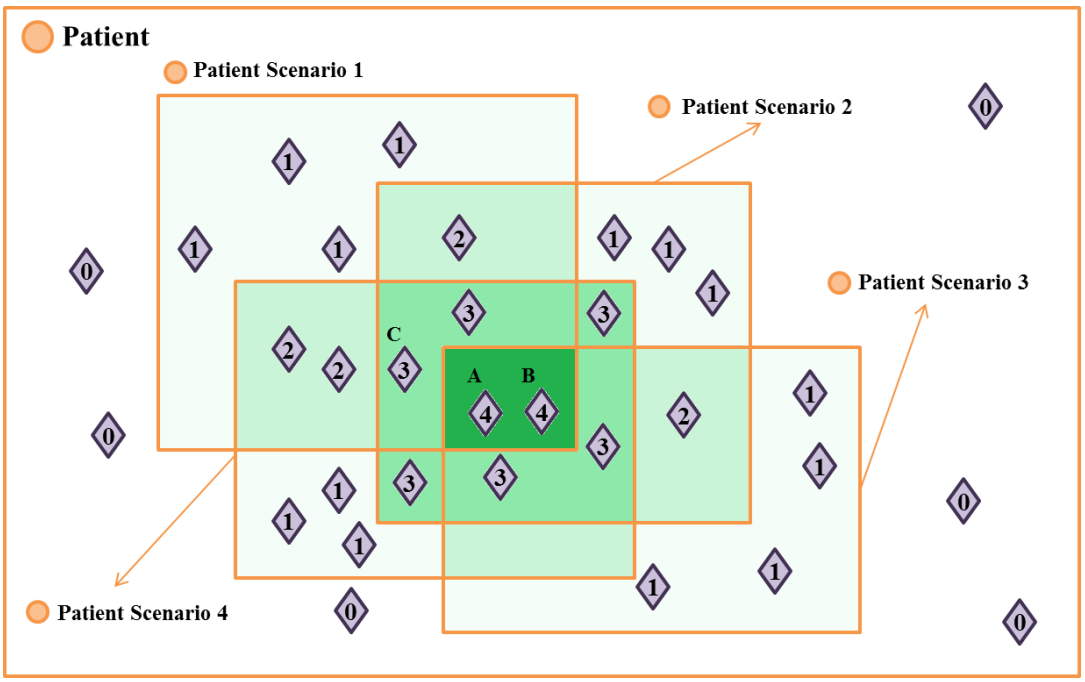

Figure 4.3: A limited ontology, where the patient individuals (purple diamonds) are subsumed by 4 different Patient Scenarios (orange boxes). As the number of parent Patient Scenarios that two patient individuals share increases, the level of semantic similarity between them increases as well. This level of similarity from low to high has been colour-coded in shades of green from light to dark.

In this situation, it is obvious that $\widehat{PS}$(A) = $\widehat{PS}$(B), which implies that Ind(A) and Ind(B) satisfy the same semantic constraints. From their set definitions, it is also clear that $\widehat{PS}$(C) $\subset$ $\widehat{PS}$(A), which implies that the semantic constraints that Ind(A) and Ind(C) jointly satisfy are a subset of the semantic constraints that Ind(A) and Ind(B) satisfy. Based on this ontological knowledge, we can deduce that A is more similar to B than it is to C (or any other patient individual depicted in Figure 4.3 for that matter).

This, in effect, can serve as the matching part of a case-based reasoning algorithm without having to define mathematical similarity functions or to calculate arbitrary weights. Formally, "case-based reasoning is the process of solving new problems by adapting previously successful solutions to similar problems" [198]. While implementing case-based reasoning, the knowledge engineer therefore has to make use of a technique to define similarity between different cases. Since identical matching between two cases becomes unlikely as the number of features that define a particular case increases, the matching technique would have to allow partial matches. The most commonly

implemented matching methods in case-based reasoning are: k-nearest neighbour, template retrieval, induction and knowledge guided induction [198].

In 2008, Austin reported that a case-based reasoning approach to decision support in colorectal cancer MDT meetings showed promise since it integrates well with the discursive group environment of an MDT and, at least to some extent, mirrors human cognitive memory recall, which plays a significant role in human decision making [18]. However, they concluded that the implementation of case-based reasoning was not feasible in their case due to lack of a sufficient number of past patients and the highly challenging task of separately building a case similarity framework that would have to take into account all patient features and formalise their relations mathematically.

Consistent with Austin's findings, our observations in the lung cancer MDT meetings also indicated that "human cased-based reasoning" was commonly used by the clinicians, a simple example being the oncologist suggesting a particular treatment option for a given patient because they had recently diagnosed and successfully treated a patient with similar features. In our case, the availability of a large number of past patient entries makes it possible to have a rich case base. Furthermore, our proposed guideline rule inference framework provides us with an intuitive way of defining semantic similarity between a reference patient and others. According to this, the similarity measure between two patients is based on ontological inference as performed by the semantic reasoner in order to determine which guideline rules a particular patient record is eligible for. As a result, as the number of the guideline rules, i.e. 'Patient Scenario' classes, in our ontology increases, the accuracy of the similarity measure improves. In this manner, when a new patient is entered, the ontological inference makes it possible to display similar patients or view summarised statistics and facts related to those 'similar' patients in the ontology, which may inform treatment decisions.

The technical aspects and the design stages in the implementation of the guideline rule inference framework introduced thus far is explained in the following section.

4.4. Implementation of the Guideline Rule Inference Framework

In this section, we describe the implementation stages of the guideline rule inference framework in the Lung Cancer Assistant. Our major design goals were for the framework to 1) be able to automatically infer rule eligibility for a patient record in the database and output related recommendation; 2) achieve this in real-time and in a scalable manner; and 3) avoid having to maintain a separate rule-base outside the ontology and rule engine.

These design goals were motivated by our literature review of existing CIG formalisms as discussed in Chapter 2. In our framework, the guideline rule knowledge within the ontology is stored as defined 'Patient Scenario' subclasses. In practice, this is an analogue of a knowledge repository of a CIG formalism, and needs to be connected with the patient records. In order to achieve this connection while satisfying our design goals, we tested different solutions which made use of the Java OWL API [85] for programmatic access to and manipulation of the LUCADA ontology.

4.4.1. Dynamic A-Box Realisation of Patient Records

While designing the LUCADA ontology, we ensured that it was capable of representing all entities in the LUCADA data model. In order to make use of ontological realisation for inferring what guideline rules apply to a patient record, we needed a program that could integrate the flat and tabular structure of the LUCADA data model with the semantically richer LUCADA ontology. From the previously discussed related work, we were aware that one way of achieving this was by representing patient records as OWL individuals in the LUCADA ontology. For this purpose, we set out to develop a Java program that could take a patient record from the database as its input and output an A-Box representation of the record in the LUCADA – SNOMED-CT ontology.

4.4.1.1. Implementation Strategy

Theoretically, the transfer of all patient records from the database to the ontology could be achieved as a one-time effort by having the intended program loop through all records in the database and creating A-Box representations for each in the ontology. This approach, in effect, would render the database obsolete once all 'data' was transferred to our ontology. Ideal as this may sound; there are well established practical issues that preclude this option, especially if one is dealing with a relational database of approximately 126,000 patient records.

When it comes to their representation of schema and data specifications, ontologies and databases differ significantly. Ontologies mix schema specification (T-Box) with real data (A-Box), whereas databases make a clear distinction between the two [199]. As a rule-of-thumb, ontologies are better suited to representing domain knowledge in software applications that require a more 'enriched' meaning –such as our CDS tools- since they allow a higher level of knowledge abstraction and expressiveness [200].

However, as a result of this higher level of abstraction and the lack of a boundary between the schema specification and data storage, ontologies are grossly inefficient in storing and managing data (A-Box) when compared to relational databases. While general purpose description logic reasoners can classify large and expressive biomedical ontologies such as SNOMED-CT, they often provide limited support in dealing with large number of instances/data. Therefore, one current practice for applications that need to process a high number of instances is to store data in a database, which interfaces with an ontology [199].

Taking this into consideration, we developed an "Ontology Worker" Java class that made use of the JDBC technology [167] to access a patient record in the database and temporarily create a representative 'Patient' individual in the LUCADA ontology with the help of the OWL API. The programmatic steps carried out by the two guideline rule inference classes: "Ontology Worker" and "Database Worker" in order to create the A-Box

representation of a patient record and carry out classification and realisation are as outlined in Figure 4.4.

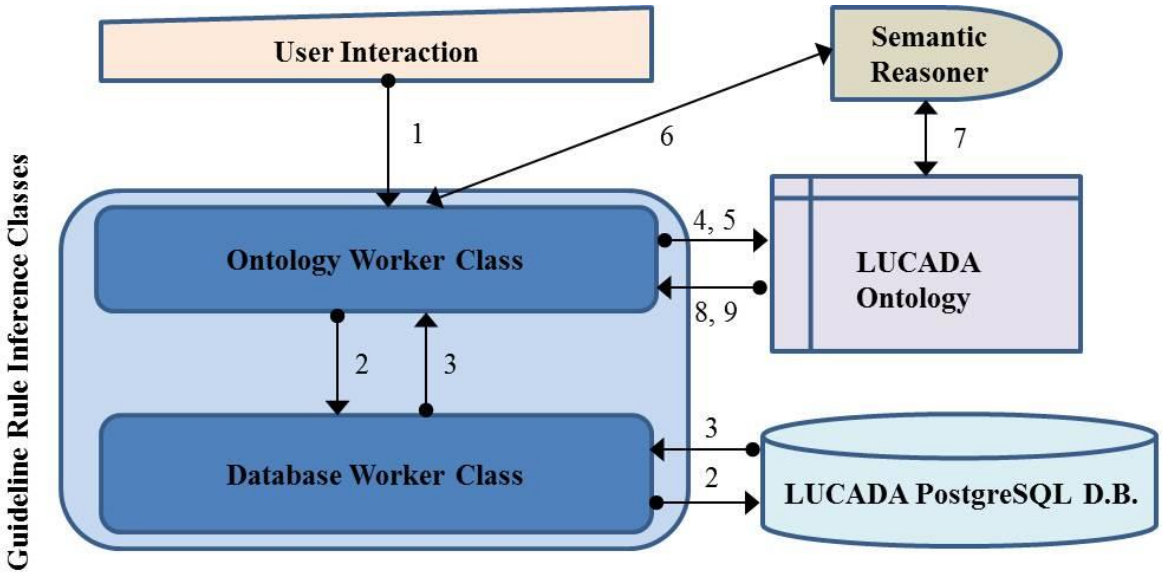

Figure 4.4: A workflow diagram that summarises the steps to achieve guideline rule inference by making use of the LUCADA Database, the LUCADA ontology, a semantic reasoner and the Ontology Worker and Database Worker Java classes that orchestrate the workflow.

1- User interaction for performing inference.

2- Query the LUCADA database to retrieve a specific patient record.

3- Commit the database query results into memory and determine which database fields of the patient record are filled in.

4- Create corresponding OWL individuals to encode the given patient record in the LUCADA ontology.

5- Fill in the object and data properties of the automatically created OWL individuals based on the database fields of the patient record (this is illustrated in Figure 4.5).

6- Initialise a semantic reasoner on the ontology.

7- Perform 'Classification' and 'Realisation' on the ontology using the semantic reasoner.

8- Return the inferred class memberships of the automatically added OWL Patient individual to determine which 'Patient Scenario' criteria it satisfies.

9- For each inferred 'Patient Scenario' parent of the OWL Patient individual, return which decisions that particular 'Patient Scenario' class supports/opposes and what those decisions entail, as given in Figure 4.2.

As a result of this process, for each Patient individual Ind(i), the 'Ontology Worker' returns the list of all applicable "Patient Scenario" classes, i.e. $\widehat{PS}$(i)={ $\widehat{PS}$: Ind(i) $\stackrel{\hat{I}}{\rightarrow}$ PS(i)}, and a list of all corresponding recommendations, i.e. $\widehat{Con}(i)$={ $\widehat{Con}$: PS(i) $\Rightarrow$ Con(i)}. Here, $\widehat{PS}$(i) is acquired via steps 7 and 8 through ontological inference on the A-Box, while $\widehat{Con}(i)$ is retrieved in step 9 by simply traversing through the predefined relations between the corresponding Patient Scenario, Decision and Treatment Plan classes.

Figure 4.5 showcases the automatically created A-Box representation of a patient record as explained in steps 4 and 5 above. It should be noted that the A-Box example shown in the figure is only illustrative and for clarity excludes many individuals and property axioms. Programmatically, the development of the code that automatically generates the A-Box representation was a very time-consuming and tedious process that required the creation of interdependent Java methods to map each database field to its A-Box representation in the ontology.

As an example, in order to formally encode the information that the patient record 'Berkan_Sesen' has a tumour on the right lung, the Java code has to automatically create not only an OWL Patient individual with the same name, but also all patient and disease specific individuals and the object and data properties between them, i.e. Berkan_Sesen $\xrightarrow{\text{hasClinicalFinding}}$ Cancer_Berkan_Sesen $\xrightarrow{\text{hasHistology}}$ Tumour_Berkan_Sesen $\xrightarrow{\text{hasLaterality}}$ Reference_Right, as depicted in Figure 4.5.

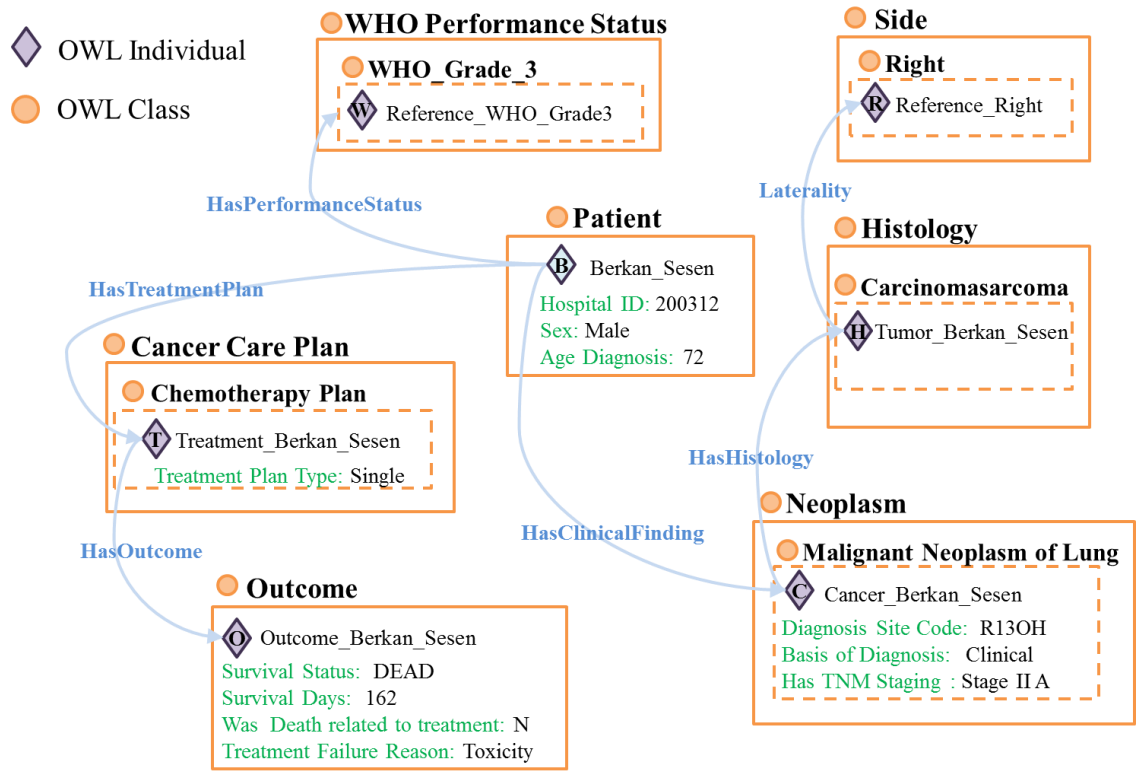

Figure 4.5: The ontological representation of a patient record in the LUCADA ontology. The blue edges stand for object properties between OWL individuals and the green list items represent the datatype properties belonging to the respective class/individual.

It should be noted that the process steps explained above do not include saving the A-Box representations to the ontology since so doing would result in an accumulation of patient records and negatively impact the performance of the inference step (step 7) as explained earlier. Instead, the A-Box representations are added to the ontology temporarily for each patient. When a new patient is to be processed, the semantic reasoner is re-synchronised to account only for the patient record being processed at the time. In the next section, we will look into the performance of this A-Box reasoning process more closely.

4.4.1.2. A-Box Reasoning Performance Results

In Chapter 2, we introduced the primary inference operations that can be carried out by a semantic reasoner and listed the most commonly used OWL-2 semantic reasoners that enable these functionalities. As explained in the previous section, our initial strategy in implementing the guideline rule inference framework relied on performing realisation on the LUCADA – SNOMED-CT ontology to infer class memberships of patient records

represented as ‘Patient’ individuals in the ontology. Since these direct class memberships are defined with respect to the class hierarchy, realisation can only be performed after classification.

In order to meet our design goals of providing decision support in a timely and scalable manner, we needed to make sure that the classification and realisation operations could be carried out in real time by selecting a semantic reasoner that was not only sufficiently fast but also sufficiently expressive to interpret the logical operators and constructors used in our ‘Patient Scenario’ equivalent class expressions. For these purposes, we carried out a series of experiments to evaluate the inference performances of the state-of-the-art OWL-2 semantic reasoners.

As discussed in the previous section, scalability of ontological inference remains one of the major obstacles to the wider adoption of ontologies in practical applications [201]. To date, various studies have benchmarked (semantic) DL reasoners [201]–[205]. Most of these studies relied on synthetic ontologies that were purpose-built for the benchmarking task [201].

In our empirical approach, we aimed to compare how the realisation times of our manually designed LUCADA ontology and the Integrated LUCADA - SNOMED-CT ontology, which we had obtained through ontology matching and automatic module extraction tools, varied as we increased the guideline rule coverage by inputting more ‘Patient Scenario’ classes into both ontologies. For this reason, we ran two sets of experiments in which we incrementally added 40 ‘Patient Scenario’ equivalent class axioms to each ontology and recorded the times taken by different reasoners to perform realisation at each step. While the addition of more ‘Patient Scenario’ classes scaled up the T-Box, we kept the A-Box constant, containing the same set of individuals that represent a patient record from the LUCADA dataset.

All 40 equivalent class expressions represented real guideline rule criteria that were taken from actual guideline documents. As a result, the complexity of the class expressions varied between different rules. Since we wanted to record the realisation times by adding one 'Patient Scenario' at a time, this variation in the complexity of different equivalence class expressions could produce non-uniform results or fluctuations between different steps. In order to eliminate this unwanted effect, we repeated all experiments 100 times, randomly permuting the order in which the 40 rules were added to both ontologies. This experimental setup is summarised algorithmically in Figure 4.6.

Experimental Setup:
Input: Ontology (O), Patient Scenario Axioms (A), Reasoner (R)
for (i=1to max_repetitions)
  $O_0 = Ontology$
  permute(A)
  for (j=0 to size(A)-1)
    $O_{j+1} = O_j + A(j)$
    Initialise (R, O_{j+1})
    $Time_{Realisation}(i, j) = R.realise(O_{j+1})$
    Dispose (R)
  end for
end for
Output: average($Time_{Realisation}$)

Figure 4.6: The experimental setup for reasoner benchmarking. This algorithm is run both with the LUCADA and Integrated SNOMED-CT LUCADA ontologies for each semantic reasoner.

Next, we looked into which semantic reasoners to include in our experiments. Two obvious groups were the general purpose OWL-2 reasoners and the OWL-2 EL profile reasoners.

1) General purpose OWL-2 reasoners. These constitute the most commonly used OWL-2 reasoners which can deal with the majority of the use cases in terms of expressivity and the range of axioms they support. From this group, we evaluated the performance of FaCT++

version 1.6.2 [82], HermiT version 1.3.7 [90] and Pellet version 2.3.0 [86], [91]. All of these reasoners have application programming interfaces that can communicate with the OWL-API.

2) OWL-2 EL profile reasoners. These operate on the OWL-2 EL, which is a restricted profile of OWL-2. We decided to include OWL-2 EL reasoners in our study since W3 consortium reports that OWL-2 EL profile is sufficient [206] to express the SNOMED-CT, on which we base the clinical terminology of our ontology. For this reason, we initially included JCel [94], and ELK 0.3.2 [95] reasoners in our study since they had readily available programmatic interfaces with the Java OWL API.

However, after running the experiments, we found that EL reasoners either failed to complete the inference process or produced warnings indicating "the reasoning may be incomplete" and consequently produced bogus results. In hindsight, we realised that this was caused by the fact that the OWL-2 EL profile did not support some axiom and restriction types that are frequently used in 'Patient Scenario' class expressions, such as the 'Object Union Of (disjunction)', 'Data Property Assertion', 'Data Has Value' axioms required by Rule 1 at the beginning of this chapter

We ran all experiments on a desktop computer with an Intel® Xeon® 2.27 GHz CPU and 15 GB of RAM. The corresponding sizes and the expressivities of the two input ontologies are as given in Table 4.1.

Ontology	Integrated LUCADA SNOMED-CT	LUCADA Ontology
DL Expressivity	ALCHIF (D)	ALCHI (D)
#Classes	1579	376
#Object Properties	63	37
#Data Properties	63	63
#Equivalent Class Axioms	1041	0
#Subclass of Axioms	999	386

Table 4.1: Comparison of the LUCADA and the Integrated LUCADA - SNOMED-CT ontologies.

The difference in the DL expressivity of the integrated ontology (ALCHIF (D)) is due to the additional functional properties that were inherited from SNOMED-CT.

The average realisation times calculated in the experimental setup (given in Figure 4.6) for the LUCADA and the Integrated LUCADA SNOMED-CT ontologies are given in Figures 4.7 and 4.8 respectively.

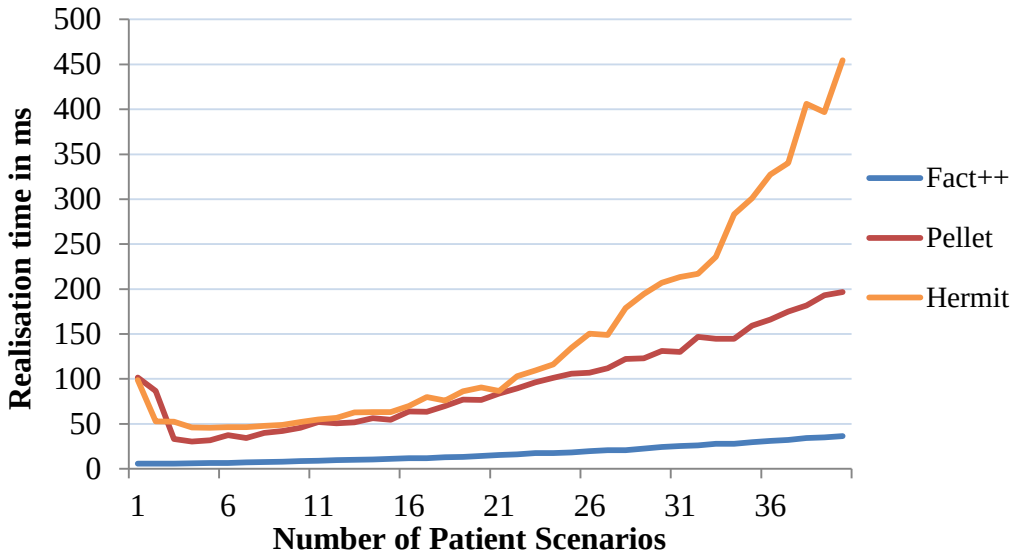

Figure 4.7: Average times taken by different reasoners to realise a single Patient individual versus the number of Patient Scenario classes in the LUCADA ontology.

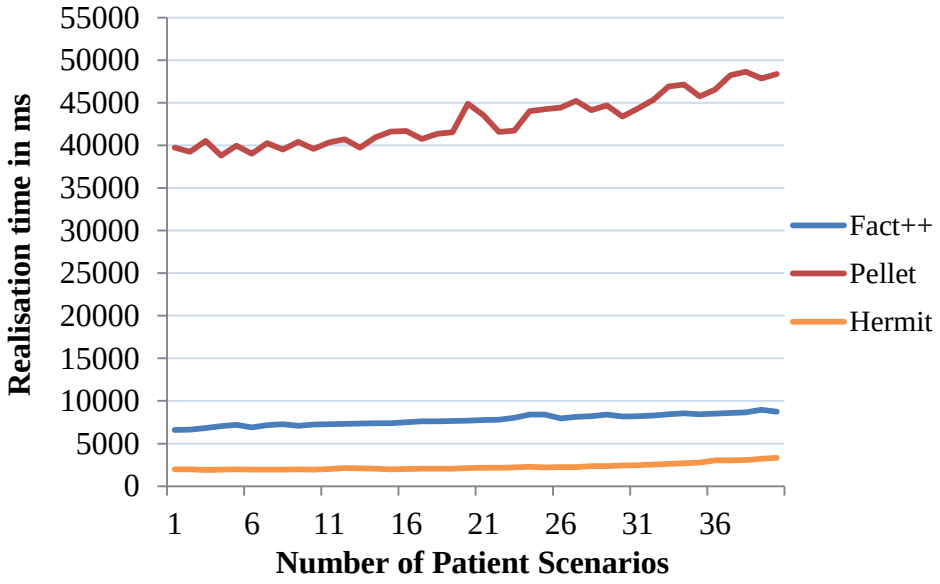

Figure 4.8: Average times taken by different reasoners to realise a single Patient individual versus the number of Patient Scenario classes in the Integrated LUCADA - SNOMED-CT ontology.

An initial observation from both plots is that the inference times for the LUCADA ontology are significantly lower than those for the Integrated LUCADA - SNOMED-CT ontology, which is not surprising due to the additional axioms that have been inherited from SNOMED-CT in the latter ontology. Furthermore, it is evident that for both ontologies, the more 'Patient Scenario' classes we add into the T-Box, the longer it takes for the semantic reasoners to realise the 'Patient' individual.

Focusing on the realisation times for the LUCADA ontology, the time scale for realisation goes only up to 0.5 seconds with the slowest reasoner for the given task, Hermit. From Figure 4.7, we can see that the two best performing reasoners are Fact++ and Pellet with seemingly linear increases in realisation times with respect to the number of 'Patient Scenario' classes. On the other hand, the realisation times for Hermit increase exponentially. This is not a good sign when one takes into account that the CDS prototype will realistically cover more than 40 rules after a more extensive knowledge elicitation.

For the Integrated LUCADA - SNOMED-CT ontology, we observe a more varied set of realisation time results for different reasoners. The best performing reasoners for this ontology are Hermit, followed by Fact++. It can be seen that for these two reasoners, the increase in realisation time with respect to the number of 'Patient Scenario' classes in the ontology seem linear for the covered interval. However, for Fact++, there is an offset of approximately 6 seconds, which is significantly higher and not compatible with our goal of producing instantaneous results. This offset in the case of Pellet is as high as 40 seconds, which is surprisingly slow. Following subsequent literature review and meetings with the semantic reasoning experts from the Oxford Computer Science department, we learned that the poor performance of Pellet in reasoning with SNOMED-CT is known, and has been reported in previous studies as well [207], [208].

These results revealed that the real-time A-Box reasoning was not a scalable implementation approach for our guideline rule inference framework. Another downside of

depending on A-Box reasoning was the necessity to re-synchronise the semantic reasoner in order for it to take into account the axiomatic modifications we made to the ontology. The re-synchronisation needed to be carried out for each patient record, subsequent to the Ontology Worker class creating the set of A-Box axioms representing a patient record, as shown in Figure 4.5. In addition, once the realisation of the OWL individual was complete, an Ontology Worker class method had to remove all A-Box axioms corresponding to the current patient record in order to avoid accumulating A-Box axioms in the ontology.

Fortunately, the re-synchronisation process was not as time-consuming as reloading the entire ontology to the semantic reasoner, since all the three general purpose OWL-2 reasoners tested support incremental reasoning [209], [210], i.e. reasoning that reuses the results obtained from previous computations. In order to investigate the additional time that would be introduced with this re-synchronisation step, we ran an additional set of experiments with the LUCADA ontology, in which we focused on recording the ontology re-synchronisation and individual realisation times separately for the LUCADA ontology with 40 'Patient Scenario' classes. We processed 100 patient records consecutively, according to the steps given in Figure 4.4 and measured the average times taken at each step to re-synchronise the ontology and realise a single individual. The results are given in Figure 4.9.

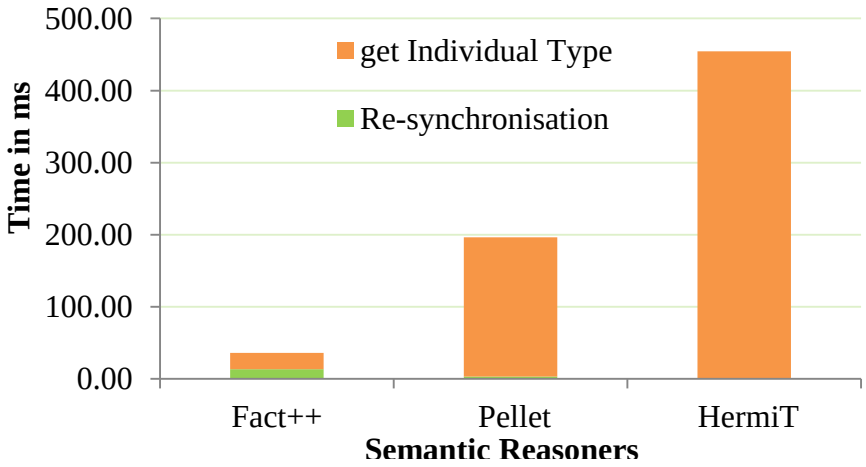

Figure 4.9: The re-synchronisation and realisation times taken by the three semantic reasoners for the LUCADA ontology with 40 'Patient Scenario' classes.

As can be seen Figure 4.9, Fact++, which is the best performing reasoner on the LUCADA ontology according to both results in figures 4.7 and 4.8, actually spends approximately a third of the overall time reported for individual realisation on re-synchronising the reasoner. On the other hand, the re-synchronisation times for Pellet and HermiT constitute only 2% and 0.03% of their total reasoning times respectively. These results indicate that Pellet and HermiT are quite fast to re-synchronise with respect to the changes but are considerably slow in getting the types of a given individual, whereas Fact++ is relatively slow in re-synchronise each step but very fast in realising a 'Patient' individual.

Overall, due to the inherent limitations of performing A-Box reasoning and having to make physical changes to the ontology file, we investigated an alternative implementation strategy that will be explained in the next section.

4.4.2. Dynamic T-Box Classification of Patient Records

After the unsatisfactory performance results of our initial attempt to communicate patient records in the database with our ontology, it became clear that we needed to avoid A-Box reasoning since it did not meet our fast response criterion. Furthermore, having to make modifications to the ontology introduced an additional step of re-synchronisation prior to realising the individual of interest. Finally, another motivation for an alternative strategy was the highly tedious and error-prone process of developing interdependent Java methods to create an A-Box representation of a patient record.

4.4.2.1. Implementation Strategy

As a result, we started looking for an alternative implementation that did not involve the drawbacks mentioned and was easier to code. At this point, we were inspired by the DL query tab within Protégé [83], which allows the user to run classification or realisation queries based on (pseudo) DL class expressions. According to this, we can represent a

patient record as a pseudo class expression (instead of a collection of OWL individuals) and return its subsumption relations by classifying the ontology.

However, writing class expressions in the functional-style OWL syntax can become a daunting task due to the verbose and complicated grammar. As an example, Figure 4.10 gives the OWL-2 class expression in the functional-style syntax for representing the same patient record depicted in Figure 4.5.

```
ObjectIntersectionOf (<#Patient>
DataHasValue(< #Sex> "Male"^^xsd:string))
DataHasValue(< #AgeDiagnosis> "72"^^xsd:integer)
ObjectSomeValuesFrom(< #hasPerformanceStatus> < #WHOPerformanceStatusGrade3>)
ObjectIntersectionOf(ObjectSomeValuesFrom(< #hasClinicalFinding> ObjectIntersectionOf(<
#MalignantNeoplasmofBronchusorLungNOS> ObjectSomeValuesFrom(< #hasPreHistology>
ObjectIntersectionOf(< #Carcinosarcoma> ObjectSomeValuesFrom(< #LATERALITY> < #Right>)))
DataHasValue(< #BasisofDiagnosis> "Clinical"^^xsd:string) DataHasValue(< #hasPreTNMStaging>
"IIA"^^xsd:string)))
ObjectSomeValuesFrom(< #hasTreatmentPlan> ObjectIntersectionOf(< #ChemotherapyPlan>
ObjectSomeValuesFrom(< #hasOutcome> ObjectIntersectionOf(< #Outcome> DataHasValue(<
#SurvivalDays> "162"^^xsd:integer) DataHasValue(< #SurvivalStatus> "DEAD"^^xsd:string)
DataHasValue(< #WasDeathRelatedtoTreatment> "Yes"^^xsd:string))) DataHasValue(<
#TreatmentPlanType> "Single Modality"^^xsd:string)))
ObjectExactCardinality(1 < #hasClinicalFinding> < #ClinicalFinding>)
)
```

Figure 4.10: The OWL-2 functional style class expression that has been parsed from the automatically created OWL DL query in Manchester syntax, given in OWL Patient DL Query

Fortunately, the Manchester OWL syntax provides a more user friendly and compact way to encode such class expressions [81]. Figure 4.11 gives the Manchester OWL syntax representation of the same patient record represented in Figure 4.10. As can be seen, this frame-based representation is a lot easier on the eye and allows the knowledge engineer to type and understand logical expressions more easily.

Motivated by this, we developed a new version of the “Ontology Worker” Java class, which could automatically translate a patient record into a Manchester OWL syntax class expression based on the patient record fields in the database. For this purpose, we prepared Manchester OWL syntax mapping templates between the database fields and the ontology properties. When database query results are input, this new version of the Ontology

Worker class automatically fills in the corresponding templates based on the patient field values retrieved from the database and appends them to a single 'StringBuffer' object, which upon completion, constitutes the 'Patient' class expression (Figure 4.11). This class expression is then parsed into the OWL functional syntax using the OWL API and a semantic reasoner is initialised to perform classification to return the super classes of the automatically created 'Patient' class expression.

```
Patient and
Sex value "Male"^^string and
AgeDiagnosis value 72
and (hasPerformanceStatus some WHOPerformanceStatusGrade3 )
and hasClinicalFinding some (MalignantNeoplasmofBronchusorLungNOS
and (hasPreTNMStaging value "IIA"^^string) and
(BasisofDiagnosis value "Clinical"^^string)  and
(hasPreHistology some (Carcinosarcoma and (LATERALITY some Right))))
and hasTreatmentPlan some (ChemotherapyPlan and
(TreatmentPlanType value "Single Modality"^^string) and
(hasOutcome some (Outcome and
(SurvivalStatus value "DEAD"^^string) and
(SurvivalDays value 162) and
(WasDeathRelatedtoTreatment value "Yes"^^string))))
and (hasClinicalFinding exactly 1 ClinicalFinding)
```

Figure 4.11: The automatically created OWL DL query for the same patient record depicted in Figure 4.5 (Section 4.4.1) in the OWL-Manchester syntax.

In contrast to our initial real-time A-Box realisation approach, here we determine eligibility by class subsumption between our automatically created 'Patient' class expression and the 'Patient Scenario' classes defined in the ontology. In other words, the patient record representation and therefore the inference are achieved at the T-Box level, which can provide faster inference [82], [86], [89]. In terms of implementation costs, developing the code for this class expression approach was also a lot easier compared to the highly interdependent Java classes we had to create for automatic A-Box representation of a patient record.

In the next section, we discuss the performance of this approach in more detail. However, before continuing, we note a practical issue with the automatic creation of these class expressions. In Figure 4.11, the last logical constraint states that the 'Patient'

"hasClinicalFinding **exactly** 1 ClinicalFinding". This expression is necessary due to the Open World Assumption (OWA) used in OWL, which entails that we cannot assume something does not exist unless it is explicitly stated so [199]. OWA is not an immediately intuitive concept for those who are more familiar with the conventional closed world assumption adopted by databases and RDF models, where it is safe to assume that something does not exist if it is not explicitly stated otherwise [211].

According to our ontology, which inherits the SNOMED-CT taxonomy, both 'Cancer' and various LUCADA co-morbidity types, such as 'Renal Failure' and 'Cardiovascular Disease', are classified under the 'Clinical Finding' class. In addition, we define the object property "hasClinicalFinding" to associate a 'Patient' individual with a 'ClinicalFinding' individual (Figure 4.1). As a consequence, an ontological patient that has co-morbidities along with lung cancer end up having multiple "hasClinicalFinding" relations with the 'Clinical Finding' class. Therefore the "hasClinicalFinding **exactly** 1 ClinicalFinding" expression in Figure 4.11 is a 'closure axiom' that asserts complete knowledge and serves to distinguish patients without co-morbidities. This is necessary to accommodate for the "has no co-morbidities" condition that appears in several guideline rules. We talk about such similar phrases and the overall knowledge elicitation process more extensively in chapter 5.

4.4.2.2. T-Box Reasoning Performance Results

In order to test whether our T-Box-query based representation of a patient record met our design criteria of scalability and fast response to the clinicians, we devised a similar experimental set-up to that described in section 4.4.1.2. Accordingly, we compared how the inference times of our manually designed LUCADA ontology and the Integrated SNOMED-CT LUCADA ontology changed as we added more 'Patient Scenario' classes into both ontologies.

We used the same set of 40 ‘Patient Scenario’ classes, which we added incrementally to our ontologies to increase the T-Box size. Furthermore, we input the same automatically created ‘Patient’ class expression to query the ontologies at each step. Similar to our previous experiments, we tackled the effect of unwanted fluctuations in inference times, which might arise from the variations in the complexity of different guideline rules, by repeating all experiments 100 times with random permutations of the order in which the 40 rules were added to both ontologies.

The major difference in this set of experiments is that since we accomplished patient representation at T-Box level, we did not have to modify the ontologies by adding a collection of OWL individuals. In other words, the automatically created (pseudo) ‘Patient’ class expressions allowed us to query the ontology and retrieve the inference results without having to actually modify the ontology file. As a result, instead of re-synchronising the reasoner every time a new patient record was processed, we could perform classification only once at system initialisation and after that we only needed to take into account the query times to return the super classes of the (pseudo) Patient class expressions.

Figures 4.12 and 4.13 give the T-Box query time results for the LUCADA and Integrated LUCADA SNOMED-CT ontologies, with respect to rule coverage, respectively. As can be seen from these figures, the T-Box query times achieved by the semantic reasoners for both ontologies are significantly lower compared to the A-Box reasoning times reported in the previous section. Focusing on Figure 4.12, we see that the query times achieved are on a time scale of 0 to 100 milliseconds. This is approximately five times faster than the individual realisation times reported in Figure 4.7.

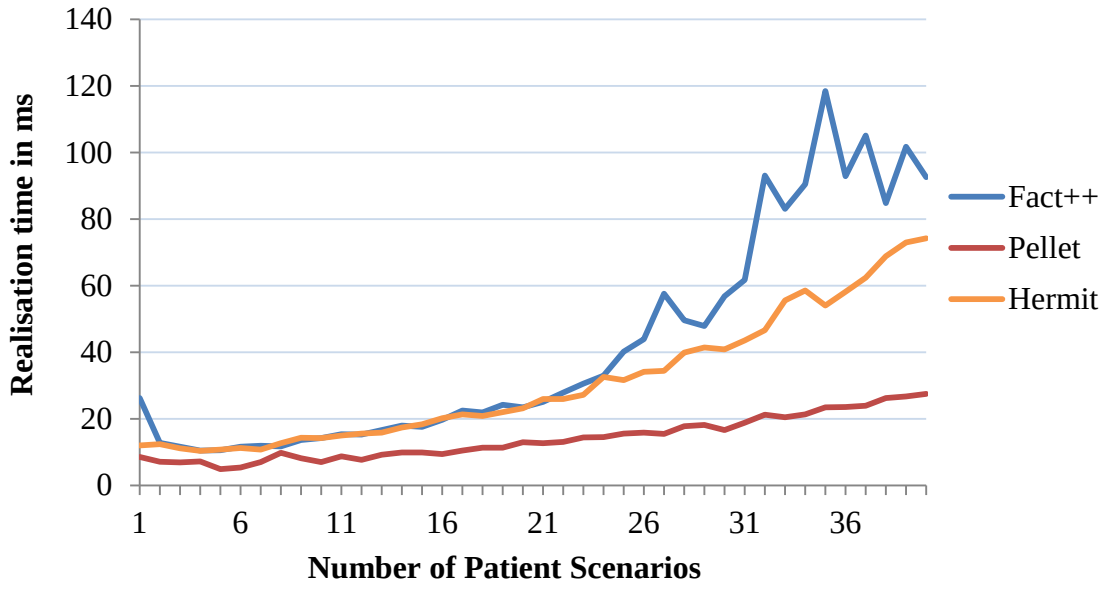

Figure 4.12: Average times taken by different reasoners to infer the super classes of a Patient class expression DL query versus the number of Patient Scenario classes in the LUCADA ontology.

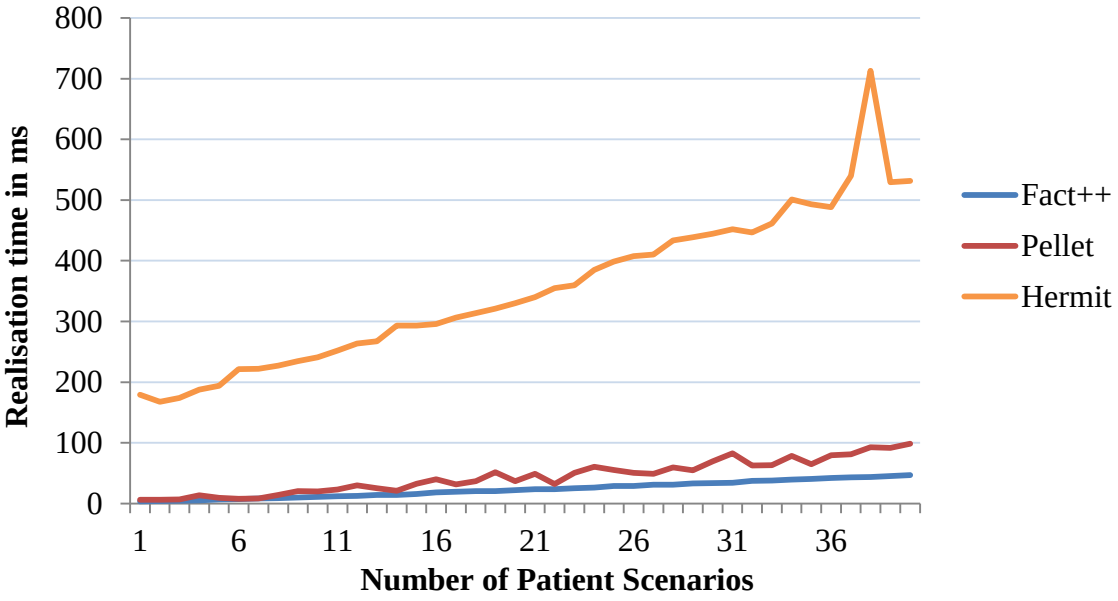

Figure 4.13: Average times taken to taken by different reasoners to infer the super classes of a Patient class expression DL query versus the number of Patient Scenario classes in the Integrated LUCADA - SNOMED-CT ontology.

However, the improvement in performance is even more drastic in the query inference times for the Integrated LUCADA - SNOMED-CT ontology. As can be seen, without having to re-classify the larger integrated ontology at each step, we have achieved inference times that are practically instantaneous even for the largest T-Box case with 40 'Patient Scenario' classes. This indicates that the greatest portion of the times recorded to perform realisation in the first set of experiments was actually allocated to re-

synchronising the ontologies (via re-classification and realisation) at each step. This observation is graphically the most explicit between the realisation times plotted for Pellet (40,000 – 50,000 milliseconds) in Figure 4.8 and the query times for Pellet (0-100 milliseconds) reported in Figure 4.13.

Overall, the query times achieved by Fact++ and Pellet are particularly encouraging since they do not exceed 100 milliseconds even for the largest T-Box with 40 scenarios. Furthermore, the increase in query times for Fact++ and Pellet with respect to the guideline rule coverage of the ontology seems to be approximately linear. If we extrapolate these performance results for an ontology with a guideline coverage of 100 rules, it is highly likely that the query times would remain between 0.5 to 1 seconds, which would provide instant feedback to the clinicians.

4.5. Discussion

According to our proposed framework, all clinical concepts (the concept of a patient, treatment option, disease and so on) and guideline-rule-knowledge are represented and reasoned upon with a semantically rich and standardised ontology. This allows representing clinical knowledge in an easily sharable, open-source format, which can also be used to draw inferences from incomplete or inconsistent sets of rules.

The strength of this approach mainly stems from making use of a highly expressive semantic language as OWL-2 to model medical knowledge based on the standardised SNOMED-CT ontology. The semantic reasoners that accompany OWL-2 enable us to interpret and infer new knowledge in real time by auto-generated DL queries that represent patient records.

To our knowledge, this purely ontology-based framework to infer patient scenario eligibility through automatically generated, patient-representative DL queries is novel. By avoiding the creation of patient individuals in the ontology, the inference times are

extremely fast and have been proven to be scalable even with a database with approximately 130,000 patient entries.

While other formalisms, such as rule engines or decision trees can be utilised to formalise guideline rules in the form “If-Then” rules, the guideline rule inference framework defined here has the distinct advantage of being able to capture the semantics of the concepts involved based on a globally acknowledged reference ontology. This is one of the main reasons why the biomedical and bioinformatics communities make use of ontologies to represent their domain knowledge [212] .

Chapter 5 - Guideline Rule-based Decision Support

In the two previous chapters, we discussed the design of the integrated LUCADA SNOMED-CT ontology and introduced our scalable and general purpose ontology-based guideline rule inference framework. In this chapter, we first present the knowledge elicitation process we have followed to create a guideline rule base in lung cancer care. In so doing, we discuss general issues in the computerisation of guideline rules and highlight some useful aspects of our guideline rule inference framework, which facilitates this process.

Following our discussion of knowledge elicitation, we introduce our online CDS prototype Lung Cancer Assistant (LCA), which uses our guideline rule inference framework to provide guideline rule-based decision support to clinicians. Then we sketch the software architecture of our online prototype, and evaluate how well it performs on predicting the recorded treatments on a patient subset taken from the LUCADA dataset.

5.1. Computerised Guideline Rules

Formalising narrative clinical guidelines involves an immense amount of effort. This is largely because ambiguity, inconsistency and incompleteness are frequently encountered in many guideline documents [213]. As a result, the task of computerising a guideline becomes much more complex than simply translating the text into machine readable expressions.

As well, in most use cases, the narrative guidelines need to be re-conceptualised to fit the knowledge base structures and modelling constraints of the application in hand [214]. In our case, these translate into 1) ensuring the conformity of the concepts used in the guideline rules with the domain coverage of our ontology and 2) limiting our expressivity to the Description Logic modelling constructs available within OWL-2.

5.1.1. Reviewing the Guideline Documents

In order to build the guideline rule base for our CDS prototype, we carried out detailed reviews of the four publicly available national and international guideline documents in lung cancer care: 1) British Thoracic Surgery (BTS) guidelines [7], 2) National Institute for Clinical Excellence (NICE) guidelines [3], 3) European Society for Medical Oncology (ESMO) guidelines [6] and 4) National Comprehensive Cancer Network (NCCN) guidelines [215].

The common function of these clinical guideline documents is to distil the current state of practice from evidence-based knowledge sources -such as clinical studies, systematic reviews and randomised trials- and present them in a more compact and structured manner. All four documents contain guideline rules for the diagnosis, treatment selection, and follow-up procedures concerning a patient's journey.

Consistent with our design goals, we limit the scope of decision support in Lung Cancer Assistant (LCA) to assisting the primary treatment selection decisions, which usually are associated with more uncertainty and practice variation, compared with other decisions on diagnosis, staging or follow-up procedures. As a result, during our review, we focused on extracting treatment selection related rules. We carried out detailed reviews of the four guideline documents listed above in the given order.

5.1.1.1. British Thoracic Society (BTS) Guidelines

The BTS guidelines are published as a joint initiative of the British Thoracic Society and the Society for Cardiothoracic Surgery in Great Britain. The most recent version that we reviewed was published in 2010 and focused entirely on lung cancer patients who could potentially be managed by radical treatment [7]. The document is broken down into sections that concern the management of specific disease subsets (e.g. T3 disease, N2

disease) and sections that set out eligibility criteria for specific radical treatment types as surgery, chemotherapy and radiotherapy.

All guideline rules are given grades, ranging from A to D, depending on the type of evidence that supports the particular rule. According to this grading system, rules graded with an 'A' are the most reliable with "*at least one meta-analysis or randomised control trial rated as 1++ and directly applicable to the target population*", while rules graded as 'D' are deemed to be the least reliable since they make use of "*extrapolated evidence from studies rated as 2++*".

5.1.1.2. National Institute for Clinical Excellence (NICE) guidelines

The 2011 version of the NICE clinical guideline document for the diagnosis and treatment of lung cancer has been put together by the National Collaborating Centre for Cancer [3]. The NICE guidelines constitute the most comprehensive set of national rules for lung cancer care in the UK.

The document covers various topics including communication, diagnosis, staging, radical treatment selection and patient follow-up strategies. In addition to these, it also provides some limited coverage on palliative care, which is offered to patients who are past curable stage, usually due to metastasis.

5.1.1.3. European Society for Medical Oncology (ESMO) guidelines

The ESMO is the leading European professional organisation that represents medical oncologists. They publish guidelines intended to provide the oncology professionals with the best standards of care. Compared to the BTS and NICE documents, ESMO guidelines are presented in a less structured format, discussing different disease subsets and treatment modalities independently and providing brief recommendations at the end of each section.

Similar to BTS guidelines, the ESMO recommendations are graded by the guideline rule generation panel with respect to the level of evidence and the level of confidence on the recommendations according to their own criteria.

5.1.1.4. National Comprehensive Cancer Network (NCCN) guidelines

In order to extend our rule coverage as much as possible, we also reviewed the American NCCN guideline documents [215] on the management of non-small cell and small cell lung cancer. We found the NCCN guideline documents by far the best structured and easiest to read among the four documents we reviewed.

Similar to their European counterparts, these documents were generated by large expert panels. However, the two major differences of the NCCN guidelines were 1) their non-linear structures, which made navigating between related sections of the document a lot easier with hyperlinks; and 2) the representation of recommendations in the form of flow charts, which leave a lot less room for ambiguity and are therefore readily convertible to machine readable format in the form of logical axioms.

5.1.1.5. Observations on Guideline Rule Coverage

As a result of our detailed reviews, we extracted 84 treatment related rules from these guideline documents. We began our review with the national BTS and NICE guidelines. Following this, we extended our search to first include European guidelines from ESMO and then American guidelines from the NCCN. The break-down of 84 treatment selection related guideline rules with respect to their source documents are given in Figure 5.1.

Guideline Document	Number of Rules
BTS	41
NICE	25
ESMO	7
NCCN	11
Total number of Rules	**84**

Table 5.1: Guideline Rule Base with respect to different source documents

5.1.1.6. Overlapping and Contradicting Rules

During the extraction of these free-text rules, we noted that there was a significant amount of overlap between different guideline documents, which was reassuring not least for cross-validating some recommendations. The overlap was particularly high between the national BTS and NICE guidelines. In Table 5.1, it can be seen that the number of rules gathered from the BTS document is higher than those extracted from the NICE document. This was because we aimed to avoid repetitions and to keep the rule base as compact as possible by not including the NICE rules that conveyed exactly the same information as some BTS rules we had already included. The lower number of rules extracted from the ESMO and NCCN documents can be explained by the same reason.

However, apart from the overlapping information, there was also discordance between the treatment recommendations in the different guideline documents. For instance, whereas the BTS guidelines report surgery as inappropriate for limited stage SCLC patients, the NICE guidelines recommend considering surgery as an option for early-stage SCLC patients, staged as T1-2a, N0, M0 [3], [7]. Another example of disagreement was the predicted post-operative (ppo) FEV1% cut-off values for post-operative dyspnoea between the two national guideline documents. The BTS guidelines indicate that patients with a ppoFEV1% value of 40 and below are in moderate to high post-operative dyspnoea risk and should be informed of the risks [7], whereas NICE guidelines identify this risk group with ppoFEV1% values of 30 and below [3].

In line with the ontological characteristics of our knowledge-base, set out in Section 4.3.1, we included all such rules in our knowledge-base, in effect allowing our rule-base to contain conflicting information.

5.1.1.7. Rule Coverage

During our review of the guideline documents, we noticed that the availability and detail of rules varied significantly between different TNM disease stages. The main criteria adopted by guideline development groups is to prioritise disease subsets that 1) have the highest level of uncertainty and 2) encompass a relatively large number of patient cases [3]. We encountered a significantly higher number of rules concerning locally advanced NSCLC patients, compared to early stage NSCLC patients. Rule coverage was particularly extensive for T3 and N2 patients. This can be explained by the fact that the decisions with the highest amount of uncertainty generally relate to locally advanced patients, who are borderline resectable, resulting in more guideline rules that concern these cases.

Another observation was that the rule coverage in the guideline documents was mostly limited to curative treatment plans. This is not very surprising since the evidence sources, on which the guidelines are based, usually adopt disease-free survival as the clinical outcome indicator and focus on curative treatments. The NICE and NCCN guideline documents have subsections, outlining the different types of palliative and supportive care (e.g. Procranial Irradiation, Psychological Support) that can be given to patients who are past curable stage. However, none of the guideline documents include rules that actually describe patient cohorts who should be offered “Active Monitoring” or “Palliative Care” as opposed to a curative treatment type.

Due to this lack of rule coverage, most guideline-based CDS tools focus on recommending curative treatments and completely exclude palliative cases [18], [26]. While this is perfectly understandable, it is not compatible with the reality of care, especially in the domain of lung cancer care, where the majority of the patients are only diagnosed at a stage where they are incurable and are therefore given palliative care. As can be seen in Figure 3.4, within the LUCADA dataset, patients who have been given Palliative Care actually constitute the highest percentage (23%).

5.1.2. Knowledge Elicitation from Experts

Extracting free-text guideline rules from the guideline sources described above was only the first step in building our guideline rule base. In order to make sense of these rules, we turned to the expertise of our clinical collaborators. This knowledge elicitation process in the creation of guideline rule repositories is a well-established bottleneck in the development of guideline-based CDS applications [18], [213].

In general, the narrative language for the guideline rules was implicit except for a few cases. As a result, we had to work closely with clinicians in explicating clinical terms such as "early stage cancer" and "patient with good performance status". In addition, these descriptive terms needed to be expressed within the LUCADA ontology, which necessitated formulating expert elicited mappings of such terms to LUCADA ontology classes and properties.

The formulation of such mappings was relatively easy when the corresponding terms were implicit, yet well agreed upon. The real challenge was to express ambiguous terms such as "fit for concurrent radiotherapy", "operable", "sufficiently ill" and "suitable for radiotherapy" using categorical or numerical values within the LUCADA data model. This was necessary in order to 'operationalise' the guideline rules in terms of available data [213]. For this purpose, we expressed such ambiguous terms with combinations of expert elicited values for ontological properties such as lung volume (FEV1), co-morbidity types and performance status.

However, in some cases, the LUCADA data model did not encompass the variables necessary to entirely capture the semantics of such ambiguous terms. For instance, according to the BTS and NICE guidelines, suitability for surgery should be determined by factors such as: risk of peri/post-operative mortality, cardiac functional capacity, lung function, and post-operative quality of life [3], [7]. Among these four, the LUCADA

dataset does not contain any information on cardiac functional capacity or post-operative quality of life. As a result, while attempting to represent terms as "resectable" or "operable" that indicate suitability for resection, we used surrogate indicators such as patient's performance status and the availability of a cardiovascular co-morbidity. Employing these approximate mappings was necessary in order to maximise our use of the LUCADA data.

Below are two examples of free-text guideline rules and their LUCADA ontology equivalent class expressions. In specifying the recommended treatment plans, we use the numbering established in Figure 3.4 in Chapter 3. The first rule, BTS26, was taken from the British Thoracic Guideline document [7]. It is an example of a clear and structured guideline rule, the eligibility criteria (rule antecedent) of which can be encoded in OWL-2 without much need for expert interpretation:

BTS26:

Text: "*Consider surgery as part of multimodality management for patients with T1-3N2 M0, non-small cell disease.*"

OWL-2: hasClinicalFinding **some** (NeoplasticDisease **and** (hasPreHistology **some** NonsmallCellCarcinoma) **and** (((hasPreTStaging **value** "1"^^string) **or** (hasPreTStaging **value** "2"^^string) **or** (hasPreTStaging **value** "3"^^string)) **and** (hasPreMStaging **value** "0"^^string) **and** (hasPreNStaging **value** "2"^^string)))

Recommended Treatment Plans: "Multi-modal surgery treatment plans: 9, 10 and 11."

However, it should be noted that clearly defined rules as BTS26 are actually very rare. The majority of the rules we encountered during knowledge elicitation included vague terms and definitions as in rule BTS87 below, which was also taken from the BTS guidelines:

BTS87:

Text: "*Offer small cell lung cancer patients unsuitable for concurrent chemoradiotherapy, sequential chemoradiotherapy.*"

OWL-2: ComorbidPatient **and** PoorPerformancePatient **and** (hasClinicalFinding **some** (NeoplasticDisease **and** (hasPreHistology **some** SmallCellCarcinoma))) **and** (FEV1AbsoluteAmount **some** double [<= 1.0])

Recommended Treatment Plan: Sequential chemoradiotherapy (7)

Opposed Treatment Plan: Concurrent chemoradiotherapy (8)

The reader will note that, inputting a rule such as BTS87 necessitates a formal clinical definition of "suitable for sequential chemotherapy but unsuitable for concurrent chemoradiotherapy". Challenged by a highly specific new definition such as this, the information engineer can adopt one of two different policies: extend the knowledge base by including a new (binary) concept: "Suitable for sequential chemoradiotherapy but not suitable for concurrent chemoradiotherapy" and construct a rule definition that directly makes use of this newly added binary concept. The highly undesirable aspect of this approach is that most rules would then end up extending the knowledge base with such highly specific qualitative concepts which eventually would result in an unmanageable, messy and haphazard domain representation. The alternative and more structured course of action would be to formalise the new concept by using existing concepts and values within the boundaries of the established domain representation. Since we aim to create a standardised and interoperable system, we adopted this latter approach and opted to formalise all concepts and rules using only the concepts and relations defined in our Integrated LUCADA SNOMED-CT ontology. As mentioned previously, we have had to define the criteria, for such vague and descriptive terms, based on our expert panel's judgement. It should be noted that a shortcoming of this approach is that the LUCADA-compatible versions of such clinical concepts are not always complete and may at times be

approximations rather than comprehensive, over-arching formalisations of the concepts. For instance, in the case of BTS87, we have chosen to represent suitability for chemoradiotherapy in terms of the "Performance Status", "Existence of a co-morbidity" and "FEV1 absolute amount" variables of the LUCADA model.

Unfortunately, vagueness and ambiguous wording in most guideline documents introduce an additional level of uncertainty while translating the rules into a machine readable format. From an informatics point of view, publishing a guideline rule that states "*Consider Treatment X for patients who are suitable for Treatment X*" does not convey any information. Although the quality of clinical guideline authoring has been improving with the help of observational studies raising awareness and urging guideline developers to avoid ambiguous wording [216]–[218], slight variations of the uninformative template given above were encountered frequently during our review of the BTS, NICE and ESMO guidelines. Compared to these three, the American NCCN guidelines were substantially more structured and clearly-worded.

This inadvertent use of ambiguity causes a huge variation in the interpretation of guideline rules during expert elicitation. As a result, the likelihood of two separate expert panels computerising the same set of rules to form an identical rule base becomes extremely low. This raises the grave danger of increasing practice variation despite apparent guideline adherence [218]. It is therefore vital for guideline developers to follow published criteria in producing effective and clear rules.

At the end of our knowledge elicitation process, we analysed the frequencies of all patient and disease-related variables that were used in rule definitions. Table 5.2 provides a list of all the variables that appear in our rule base, along with the number of rules they appear in. As can be seen from this table, the histology type of the tumour is the most commonly used feature, appearing in 27 guideline rules. Histology is closely followed by the T, N and M stages of the cancer. While most rules refer to these stage descriptors separately, the

summary ‘TNM stage’ indicator still appears in 18 rule definitions as a more compact and general definition for the state of the disease.

LUCADA Variables	Number of Rules that include the variable
Age	2
Performance Status	19
T Stage	23
N Stage	26
M Stage	26
TNM Stage	17
TNM Staging Version	18
Site Specific Staging Class	5
Histology	27
Tumour Laterality	2
Primary Diagnosis	1
FEV1%	6
FEV1 Absolute Amount	12
Excision Margin	2
Co-morbidities	11

Table 5.2: A list of all LUCADA concepts that appear in our rule-base, along with the number of rules they appear in.

An interesting entry in this table is the ‘TNM Stage version’, which specifies the version of cancer staging used in the rule criteria. In such rules, since the LUCADA dataset contains patient records staged with version 6 as well as version 7, we had to separately encode the guideline rule with respect to both staging versions.

5.1.3. Advantages of Our Guideline Rule Inference Framework

After the knowledge elicitation process, eligibility criteria for 84 guideline rules were added in to our Integrated LUCADA SNOMED-CT ontology in the form of defined ‘Patient Scenario’ classes. The consequents of the rules, i.e. recommended actions, were encoded as collections of binary relations as explained in section 4.2, altogether resulting in 220 arguments that support or oppose specific treatment plans.

Making use of the ontological guideline rule inference framework that we introduced in the previous chapter facilitated our rule encoding in various ways. First, the ability to make use of the SNOMED-CT taxonomy enabled us to author over-arching rules based on

different disease types and patient subgroups. As an example, in rule BTS26, we indicate that the rule is applicable to cancers with "NonsmallCellCarcinoma" histology types without having to specify individually what those sub-types are or having to encode the same rule criteria for all those sub-types individually. This is only possible since the semantic reasoners can automatically infer the hierarchical class structure in the ontology.

Another convenient aspect of the ontological encoding of rules was that we could enrich the SNOMED-CT domain knowledge by introducing defined OWL classes that captured commonly encountered clinical terms in guideline rules. As a matter of fact, we made use of two such defined classes, namely "ComorbidPatient" and "PoorPerformancePatient", in the OWL-2 class equivalence axiom for the rule BTS87 in the previous section. The definitions of these two classes are as given below:

ComorbidPatient: Patient and (hasClinicalFinding some (Dementia or DisorderofCardiovascularSystem or ExcessiveWeightLoss or OtherDiseaseInjury or RenalFailureSyndrome))

PoorPerformancePatient: Patient and hasPerformanceStatus some (WHOPerformanceStatusGrade2 or WHOPerformanceStatusGrade3 or WHOPerformanceStatusGrade4)

As can be seen, the "ComorbidPatient" class represents a patient record which has one or more of the co-morbidity types as defined in the LUCADA data model. On the other hand, a "PoorPerformancePatient" represents cases that have a performance status of 2, 3, or 4 according to the World Health Organisation grading. Other examples of such defined OWL classes that we created in authoring rule eligibility criteria are: "GoodPerformancePatient", "LimitedStageSCLCPatient" and "EarlyStageNSCLCPatient". These all serve the common purpose of semantically representing expert knowledge and eliminating the need for repetitions.

Finally, encoding all domain knowledge in an ontology also allowed us to perform consistency checks to ensure that our domain knowledge does not include any contradictory facts [86]. Overall, making use of these convenient features of ontological knowledge representation saved us a substantial amount of time in rule authoring and resulted in a much more compact and human-readable rule-base.

While these benefits may sound elementary, even trivial, to a knowledge engineer, who is familiar in ontological design, they should not be taken for granted since many CIG languages, such as Proforma and Asbru, still do not support such high level ontological knowledge representations [48] and therefore cannot utilise any of the benefits described in this section.

5.2. Lung Cancer Assistant version 1

We integrated the LUCADA - SNOMED-CT ontology that consisted of our rule base and the guideline rule inference framework, introduced in Chapter 4, to develop the first version of the Lung Cancer Assistant CDS prototype. In this prototype, we focused on providing guideline-based decision support in a way that is similar to previous research [18], [26]. We first introduce the system architecture, then discuss the realisation of the design and finally evaluate the performance of the system in predicting the recorded treatment plans in the LUCADA dataset.

5.2.1. System Architecture

The architecture of the first version of Lung Cancer Assistant was informed by our aim to develop a CDS prototype which 1) is secure, yet easy to access, 2) can provide instantaneous evidence-based decision support at the point of care, and 3) prioritises the standardisation of domain knowledge and interoperability with other software. As discussed in the previous chapter, our guideline rule inference framework satisfied the

latter two design goals. In order to make the prototype easily accessible yet secure, we decided to develop it as a web-based application.

For this purpose, we made use of the freely available Google Web Toolkit (GWT) software development kit version 2.4.0 [219] in Java. GWT allows coding asynchronous JavaScript and XML (AJAX) applications entirely in Java and handles the compilation of the client-side code into optimised JavaScript automatically during application deployment. The client-side and server-side communication is achieved by the GWT Remote Procedure Call (RPC) framework, which is based on the Java servlet architecture [220].

The GWT-based architecture of Lung Cancer Assistant (LCA) is shown in Figure 5.1. According to this, the user interacts with the CDS application through a web-based form. End user requests, such as creating, updating or saving a patient record, are communicated from the client to the server in the form of remote procedure calls.

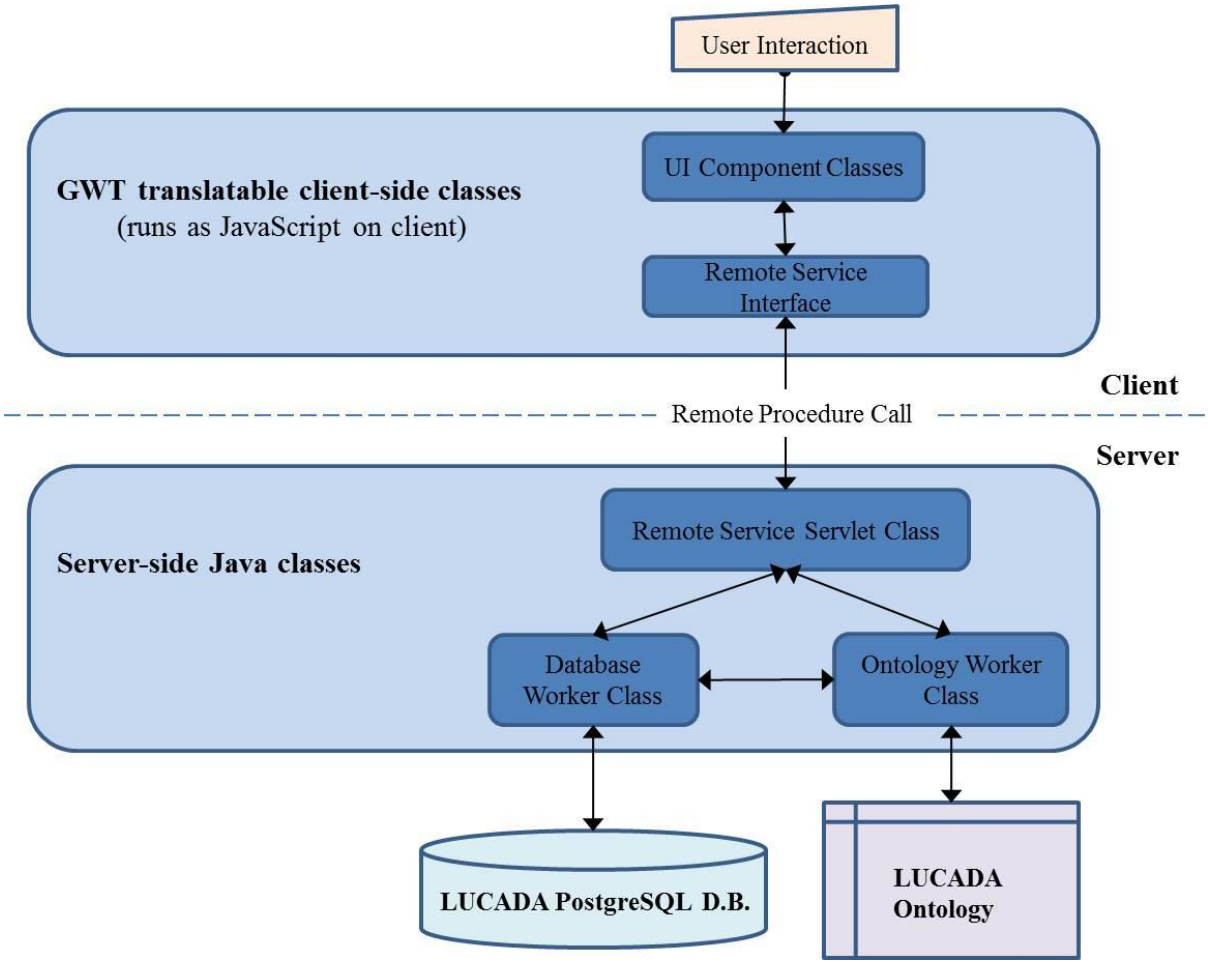

Figure 5.1: The architecture of the GWT-based Lung Cancer Assistant version 1 CDS prototype.

Depending on the nature of the end user request, the 'implemented' Remote Service Servlet class methods make use of the 'Ontology Worker' and 'Database Worker' classes in order to modify or query the database and the domain-specific ontology. The ontology is stored in OWL-2 and the patient records are stored in a PostgreSQL database. The 'Database Worker' uses JDBC [167] for connecting to and querying or modifying the LUCADA patient records, whereas the 'Ontology Worker' class utilises mainly the OWL API [85] for communicating with and querying the LUCADA ontology with the help of a semantic reasoner as described in Section 4.4.2.

5.2.2. Realisation of Design

We deployed LCA on an Apache Tomcat version 6.0 [221] server that we set up on a departmental computer. We made use of JDBC realm-based authentication [222], which is available with Tomcat, to build a secure log in page for our application and served it behind the Department of Engineering Science firewall.

This initial version of LCA allowed creating, updating and saving patient records based on the LUCADA data model. Following any of these user events, on the server side the "Database Worker" class makes the necessary changes to the database. Based on these changes, the "Ontology Worker" class runs an automatically generated DL query that represents the modified patient record and returns the patient-specific arguments for or against specific treatment plan options. Furthermore, the rule eligibility information for all patient records are kept in a separate database table, which is queried right after the DL query results are returned, in order to provide a list of similar patients as explained in section 4.3.2.

Similar to previous MDT meeting CDS prototypes [18], [26], we chose a tab-based representation for our user interface in order to provide flexibility and facilitate navigation

between different data fields. As described in chapter 1, the user interface of LCA consists of five tabs which separates the patient and disease-related data fields with respect to the subsections of the LUCADA data model, as described in [16].

Guideline rule-based decision support is presented under a separate tab. A screen shot of this "Decision Support" tab is given in Figure 5.2. Here, all 11 LUCADA treatment plan options are listed with patient-specific arguments that support or oppose them. The supporting arguments are symbolised with green "thumbs up" icons and the opposing arguments are presented with red "thumbs down" icons. In addition to this, patient-specific warnings that are neither against nor in support of a treatment plan option are also displayed as yellow triangles as can be seen in Figure 5.2.

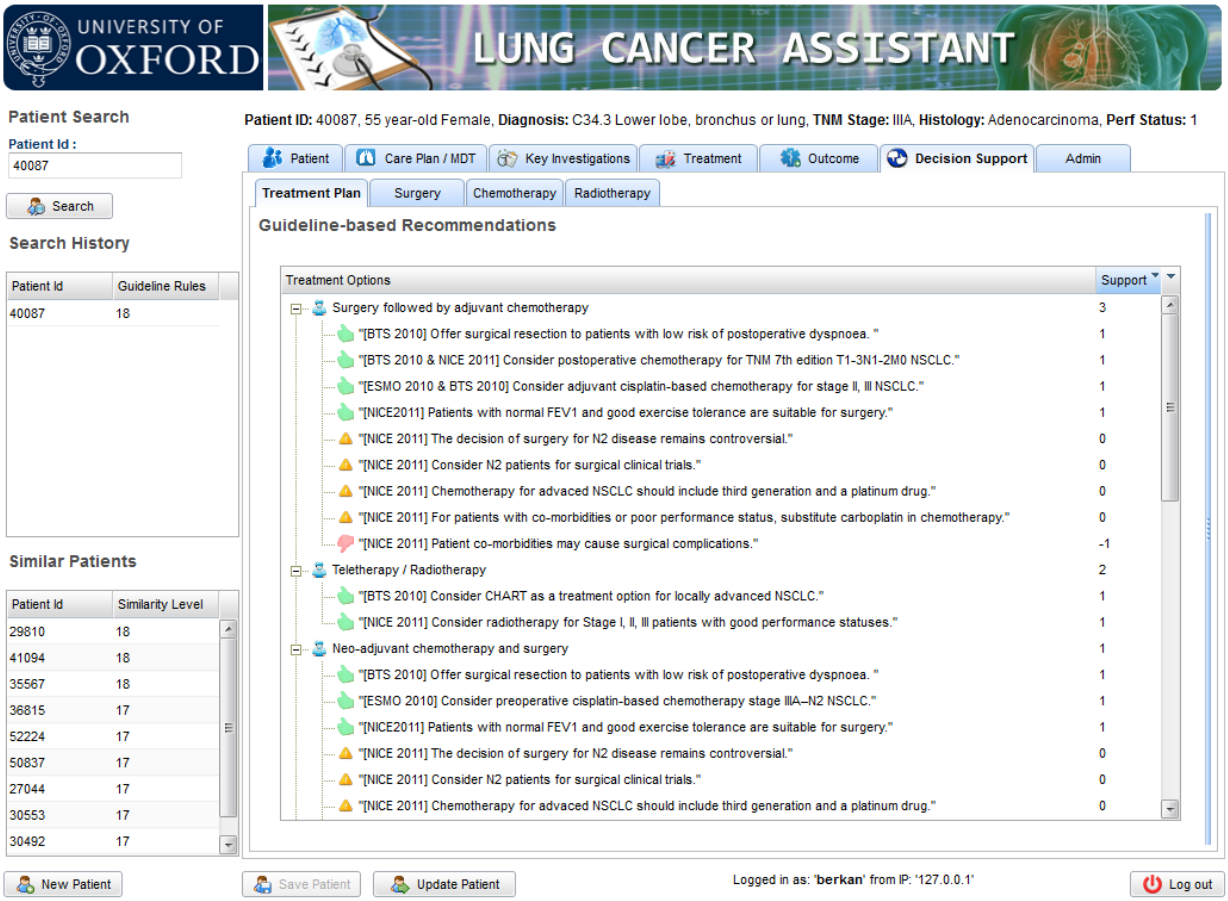

Figure 5.2: The decision support tab of LCA version 1.

Overall, for each patient record, the treatment plan that has the most support is displayed as the system recommendation at the top, with the other treatment options listed in descending order of the system's support. In this version of LCA, all supportive arguments

contribute ‘+1’ and the opposing arguments contribute ‘-1’ to the overall support of a particular treatment plan. In chapter 7, we will present preliminary work on a more structured and probabilistically sound methodology to replace this overly simplistic way of calculating net support for a decision candidate.

All arguments and warnings are presented with one-line rule summaries in the treatment plan list. However, more detailed explanations, as directly quoted from the source guideline document, pop up when the user hovers the mouse over the respective argument.

In Figure 5.2, we can see a brief patient summary above the tab-based web form. The top system recommendation for this patient is reported as “Surgery followed by adjuvant chemotherapy” with a net support of ‘+3’. This is followed by “Radiotherapy” with a net support of ‘+2’. On the search history list, we can see that 18 guideline rules in our rule base apply to this patient. On the bottom left corner, similar patients, who share common characteristics with the currently viewed patient record, are listed. The similarity level for these records is quantified based on the number of applicable guideline rules they share with the displayed patient record, as explained in section 4.3.2.

5.2.3. Guideline-based Prototype Evaluation

Once the guideline rule elicitation and the development of the first version of our CDS prototype was complete, we evaluated the performance of LCA in making plausible patient specific recommendations based on our domain knowledge, which now also included the evidence-based practice guidelines. For this purpose, we compared the top system recommendations of LCA to the treatment plan decisions actually recorded for a subset of patients taken from the LUCADA dataset. These recorded treatments, in effect, served as a “silver standard” against which we evaluated the suitability of the system recommendations.

5.2.3.1. Test Patient Selection Criteria

We applied a number of criteria to select the subset of patients from the LUCADA dataset to include in our experiment. The criteria listed below were motivated by our effort to provide unbiased results that were representative of the entire dataset, while taking into account some limitations of guideline-based decision support.

1) As mentioned earlier, the national and international guideline rule documents we reviewed mainly covered rules that applied to patients who were eligible for curative treatments. Therefore in our system evaluation, we only included patients who were prescribed a treatment plan with 'curative' intent, excluding, for example, palliative care.
2) Since our silver standard was the "Suggested Cancer Treatment Plan" as recorded in the LUCADA dataset, we only included patients for whom this field was complete.
3) In order to eliminate unwanted artefacts caused by data incompleteness and achieve best performance results by the system, we only included patients, whose most commonly encountered guideline-related fields were fully observed. As listed in Table 5.2, this translated into selecting a patient set which did not have any missing values for the top 8 most common fields in the rule-base: "Histology", "T Category", "N Category", "M Category", "TNM Category", "Performance Status", "FEV1 Absolute Amount" and "FEV1 Percentage".
4) Since mesothelioma patients were out of the scope of the guideline documents, we only included small cell and non-small cell lung cancer patients in system evaluation.

Upon retrieval of all patients that satisfied these criteria, we acquired a set of 4020 patients out of approximately 127,000 patient records that were available in the LUCADA dataset. Figure 5.3 gives a breakdown of these patients with respect to their TNM stages.

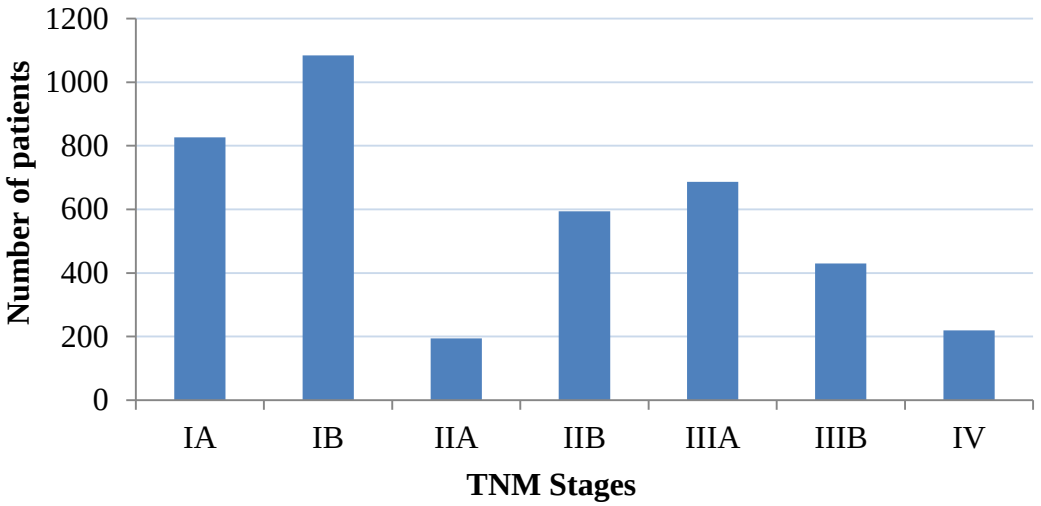

Figure 5.3: A breakdown of TNM stages of the 4020 patients included in the guideline-based system evaluation experiment

As can be seen, the number of patients with stage IV disease is relatively low, compared to the population priors given in Figure 3.1. This reflects the fact that most stage IV patients are treated with palliative care and were excluded from our experiment. Of the 4020 patient records, which were given curative treatment plans, the recorded treatments were distributed as shown in Figure 5.4.

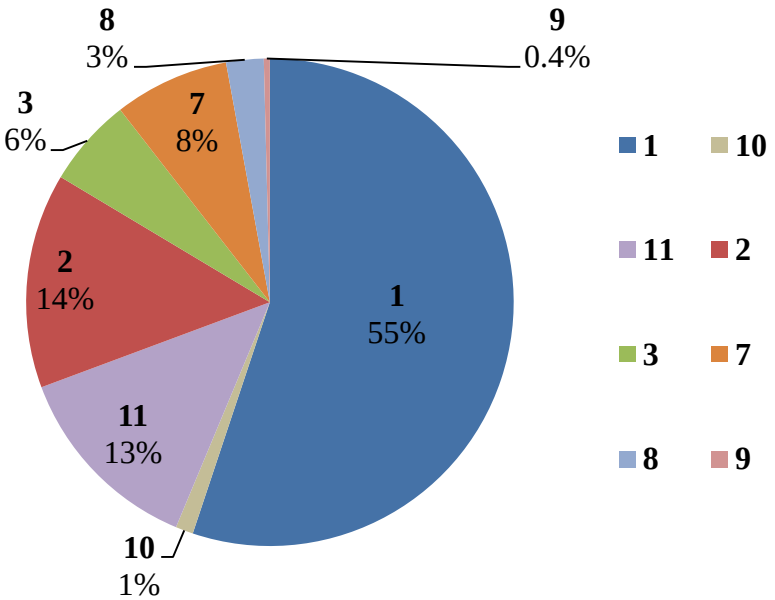

Figure 5.4: Suggested cancer treatment plan distributions in the system evaluation patient set. The suggested treatment plan codes are as listed in Table 3.3 in chapter 3.

Figure 5.4 reveals that by far the most common treatment plan type given to the experimental patient subset was Surgery (1) with 55%, followed by Radiotherapy (2) with 14% and Adjuvant Chemotherapy after Surgery (11) with 13%. The figure does not

include non-curative treatment plans “Palliative Care” and “Active Monitoring” since they were excluded by our patient selection criteria.

5.2.3.2. Evidence-based Concordance Results

Having selected our experimental patient subset, we developed a “System Evaluation” Java class on the server side in order to automatically run the experimental patient subset through LCA and evaluate concordances between the system recommendations and the recorded treatment plans. Given a list of patient IDs, the “System Evaluation” class loops through all patient records in the database and prints out the top 3 system recommendations along with the recorded treatment. In addition, it prints a confusion matrix to better visualise the commonly encountered patterns of disagreement.

Based on the system outputs, we evaluated concordance with respect to both exact and partial matches between the top system recommendation and the recorded treatment. Partial matches contained patients for whom the top recommendation of the system either subsumed or overlapped with the recorded treatment. For instance, a commonly occurring partial match pattern consisted of patients for which the recorded treatment plan “Surgery” was subsumed by the top system recommendation “Surgery followed by adjuvant chemotherapy”. Overall, the percentage of patients for whom there was an exact concordance between the top LCA guideline-based recommendation and the recorded treatment was 57%. This percentage rose to 79% when we included partial matches between the two.

Concordances with respect to recorded cancer treatment plans

We analysed the level of exact and partial concordances with respect to the recorded treatment plan types in the dataset. Figure 5.5 shows the confusion matrix summarising the aggregated discrepancies between the recorded treatment plans in the dataset and the top recommendations of LCA. The numbers on the diagonals, colour-coded in green, indicate

concordances for each treatment plan type, whereas all values that are off-diagonal represent disagreements. The orange cells, in particular, represent the most prevalent sources of discordance between the recorded treatments and the top LCA recommendations.

		Recommended Plan							
		1.Surgery	2.Radiotherapy	3.Chemotherapy	7.Sequential chemo-radio	8.Concurrent chemo-radio	9.Induction chemo and surgery	10.Neo-adjuvant chemo and surgery	11.Surgery and adjuvant chemo
Recorded Plan	1.Surgery	1639	116	49	35	138	0	0	236
	2.Radiotherapy	150	274	27	21	28	0	0	73
	3.Chemotherapy	22	24	58	9	94	0	0	30
	7.Sequential chemo-radio	33	27	31	21	146	0	0	49
	8.Concurrent chemo-radio	14	9	5	1	49	0	0	22
	9.Induction chemo and surgery	1	0	1	1	7	0	0	6
	10.Neo-adjuvant chemo and surgery	11	0	0	1	14	0	0	21
	11.Surgery and adjuvant chemo	201	11	8	5	49	0	0	250

Figure 5.5: The confusion matrix which displays the recorded treatment plans in the database versus the top guideline rule-based recommendations by LCA.

Focusing on the 'Surgery' row in Figure 5.5, we can see that while the majority of LCA recommendations are in agreement with the recorded surgery treatment plan, the discordances mainly arise due to LCA recommending adjuvant chemotherapy after surgery, whereas the recorded treatment is surgery alone. The source of this discordance is mainly the locally advanced stage patient group, which we will analyse further below. On the flip side, if we focus on the 'Adjuvant chemotherapy after surgery' row, we see that the majority of discordances (201 patients) originate from the system suggesting surgery alone. These 201 patients are all early stage patients, and the disagreement of the system stems from a guideline rule stating "*There is no evidence of benefit of postoperative*

chemotherapy in stage IA non-small cell lung cancer in a western population." taken from the BTS guidelines [7].

In addition to the confusion matrix for exact concordances, in Figure 5.6 we also give the exact and partial concordance percentages broken down with respect to the "Suggested Cancer Treatment Plan" field values of the system evaluation patient subset.

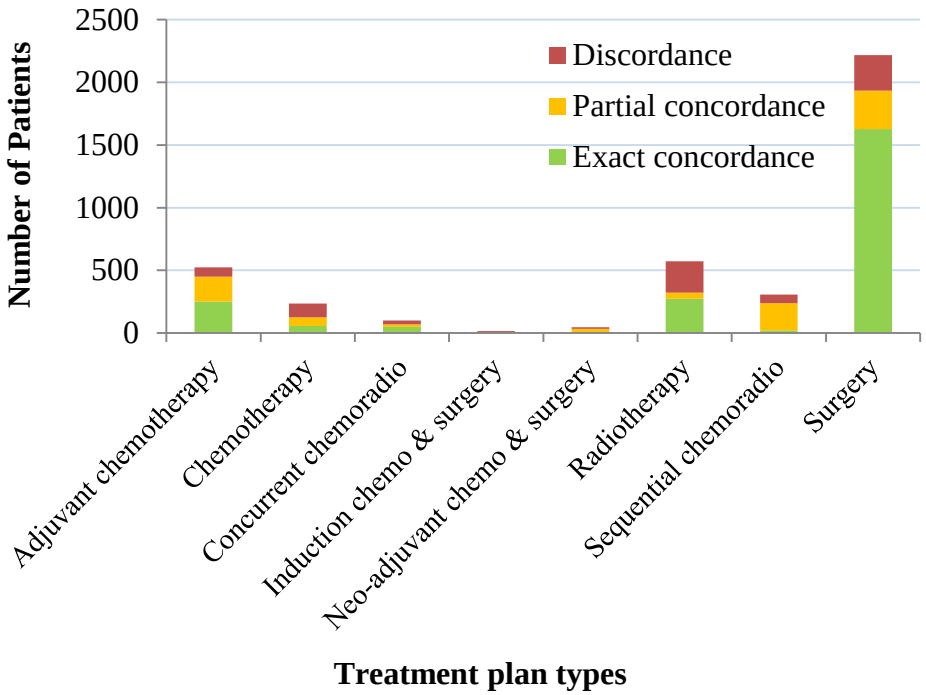

Figure 5.6: The exact and partial concordances between the guideline-based recommendations and the recorded treatment plans, stratified with respect to treatment plans.

An important pattern that is visible in both Figures 5.5 and 5.6 is that the discordant portions of the "Surgery" and "Radiotherapy" columns are mainly comprised of patients for whom LCA recommended a treatment plan involving surgery, whereas the recorded treatment was radiotherapy or vice versa. These therefore may potentially represent the more complex cases for which suitability for surgery cannot be determined by the national and international guideline rules and data recorded in LUCADA.

Similar low exact concordance percentages are also observable for patients who have been treated with "Chemotherapy" and "Sequential Chemo-radiotherapy". For the "Chemotherapy" group, as can be seen in Figure 5.5 (row 3), LCA highly favours

multimodality treatments that include- instead of being limited to- “Chemotherapy”, such as “Concurrent Chemo-radiotherapy” or “Adjuvant Chemotherapy after Surgery” treatment plans. For patients who have been given “Sequential Chemo-radiotherapy”, the LCA rule base again favours either “Concurrent Chemo-radiotherapy” or “Adjuvant Chemotherapy after Surgery” treatment plans.

As stated by the rule BTS87 given in section 5.1.3, concurrent chemo-radiotherapy is more efficacious and should be preferred to sequential chemo-radiotherapy if the patient is fit enough. Consequentially, as visible in Figure 5.5, LCA favours concurrent chemo-radiotherapy over sequential therapy, and so it may be overprescribing this treatment plan with respect to recorded clinical practice. The uncertainty here arises from the lack of clearly defined rule criteria which distinguish patients who are eligible for sequential chemo-radiotherapy from those who are eligible for concurrent chemo-radiotherapy. Therefore the discordance in these cases may be regarded as a typical example of the effects of ambiguous wording in guideline rules that are prone to varying interpretations by different expert panels.

In Figures 5.5 and 5.6, two exceptional patient groups, for whom concordance levels are also close to zero, are those who have been given “Induction Chemotherapy before Surgery” or “Neo-adjuvant Chemotherapy before Surgery”. The discrepancies between system recommendations and the recorded treatment plans for these two groups arise from the fact that no guideline rules that recommend these two treatment plans exist as of yet. Further discussions with the NLCA clinical lead, Dr Michael Peake, revealed that all these patients are enrolled in clinical trials. This is a commonly encountered pattern in clinical practice, whereby day-to-day clinical practice usually lags behind the state of the art treatment modalities due to the fact that sufficient evidence, in the form of clinical trials and peer-reviewed meta-analyses, needs to accumulate before any novel technique becomes common practice.

Concordances with respect to TNM stages

In addition to our analyses based on recorded treatment plans, we also analysed the level of exact and partial concordances with respect to the TNM stages of the test patients. Figure 5.7 shows that the concordance rates between the silver standard and top system recommendation for early stage cancer patients (Stage IA – IIB) are relatively high. This can be explained by the limited variation between the disease specifics of early stage cancer patients and their corresponding treatment decisions.

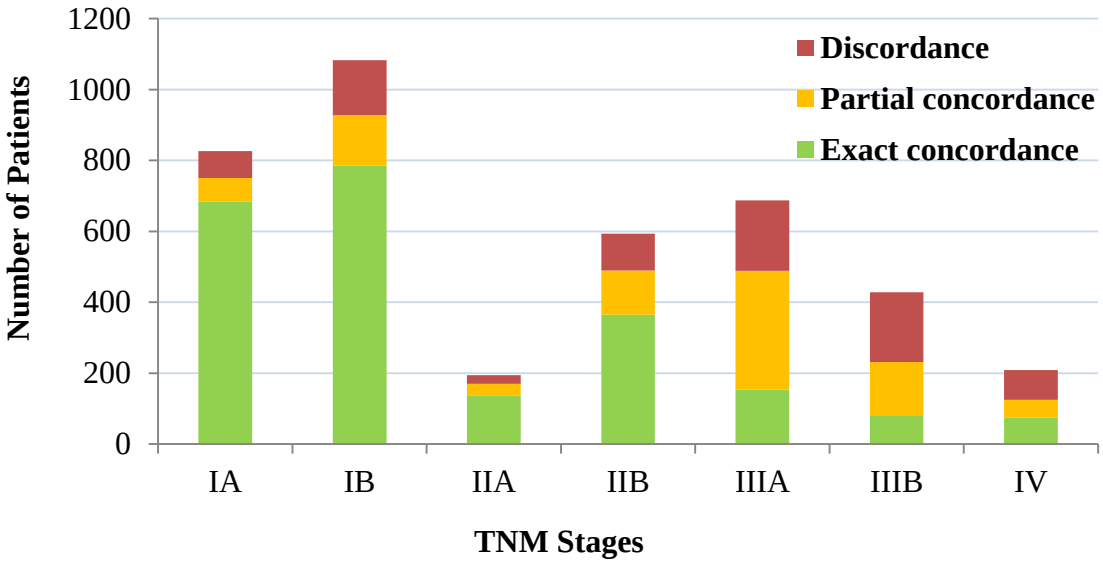

Figure 5.7: The exact and partial concordances between the guideline rule-based recommendations and the recorded treatment plans stratified with respect to the TNM stages.

On the other hand, concordance rates for locally advanced stage patients (Stage IIIA and IIIB) are significantly lower. This is not surprising since, for various reasons, locally advanced stage patients constitute the patient group with the highest degree of uncertainty. First, as can be observed in Tables 1.1 and 1.2 in Chapter 1, stage IIIA and IIIB comprise the widest range of T (tumour size and location) and N (degree of regional lymphatic node involvement) stage combinations among all TNM stages. As a result, there is substantial variation in disease specifics and, thus, treatment decisions for these patient subgroups. In addition, patients with locally advanced disease are also termed 'border-line resectable' patients for which the major source of uncertainty becomes the judgement of the MDT on

the suitability of a patient for surgery. Unfortunately, as mentioned in section 5.1.3, determining suitability for surgery is a complex decision and potentially requires additional information that is not stored in the LUCADA dataset.

One way to interpret the low exact concordance rates for the locally advanced stage patients, shown in Figure 5.7, is that despite the more comprehensive rule coverage for these patients, the national and international guideline rules are not sufficient on their own to attain high levels of agreement between LCA recommendations and clinical practice. However, it should also be kept in mind that the silver standards, against which we compare our system recommendations, do not necessarily represent best practice patterns. Therefore, the relatively low concordance rates need not necessarily indicate deficiencies of our rule base. These can alternatively be interpreted as complex cases, which deviate from best practice recommended in the national and international guideline documents.

In order to pinpoint the recorded treatment plan types for which exact concordance between the recorded treatments and the system recommendations were the lowest, we further investigated the distributions of the recorded treatment plan types for Stage IIIA and IIIB patients. Figure 5.8 summarises the exact concordance and discordance percentages of the locally advanced stage patients with respect to the recorded treatment plan types.

As can be seen in Figure 5.8, exact concordance for locally advanced stage patients who have been treated with surgery is close to zero. As previously noted during our analysis of Figure 5.5, this is mainly caused by the fact that all guideline documents recommend locally advanced stage patients who are eligible for surgery, to be given adjuvant chemotherapy after their surgery. As a result, in line with the guideline rules, LCA never recommends surgery on its own for these patients.

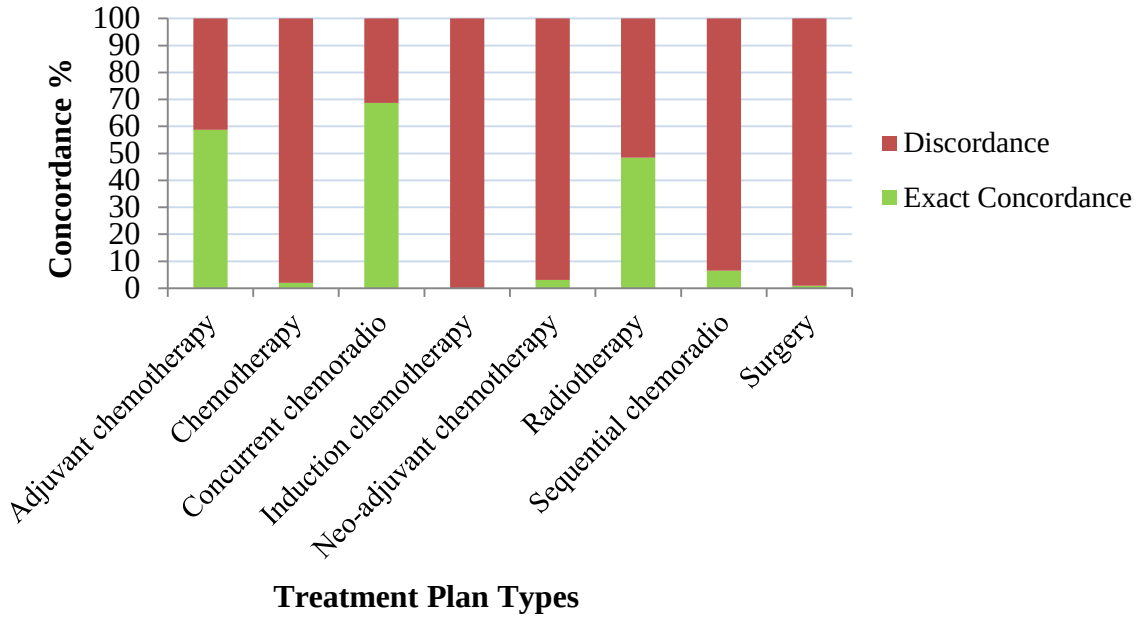

Figure 5.8: Exact concordance and discordance percentages for locally advanced (IIIA and IIIB) stage lung cancer patients with respect to their recommended treatment plan types in the LUCADA dataset.

The low level of exact concordance for patients who have been given "Chemotherapy" originates from LCA favouring multimodality chemotherapy plans over single chemotherapy. The same pattern of low exact but high partial concordances is also evident with "Sequential chemo-radiotherapy" patients. In addition, the lack of exact concordances for the "Induction Chemotherapy before Surgery" and "Neo-adjuvant Chemotherapy before Surgery" is due to the lack of evidence supporting these treatment plans. It may also be observed from Figure 5.5 that these two treatment plans are indeed offered to a very limited number of patients, who were eligible for and consented to partake in clinical trials.

5.2.3.3. The Effect of Missing Data on Performance

The deterministic guideline rule-based CDS tools, on which we based the design of the first version of LCA [18], [26], depend strongly on the assumption that all necessary data for evaluating the full set of guideline rules is available at the point of care. However, in real-world clinical datasets and electronic health records, missing data is an inevitable and commonplace artefact that simply cannot be ignored [223], [224]. For example, we have found that only 48,353 out of the 126,896 patient records (approximately 38%) in the

LUCADA dataset had a fully observed subset of the top 8 most commonly used data fields in our rule base. We return to this issue in Chapter 6.

In order to complement our results by taking into account the effect of missing data on the system performance, we carried out an additional set of experiments, whereby we contaminated the 4020-strong patient subset by methodically removing the "Histology", "TNM", "T", "N", "M" and "Performance Status" data fields, which were frequently used in determining guideline rule eligibility criteria.

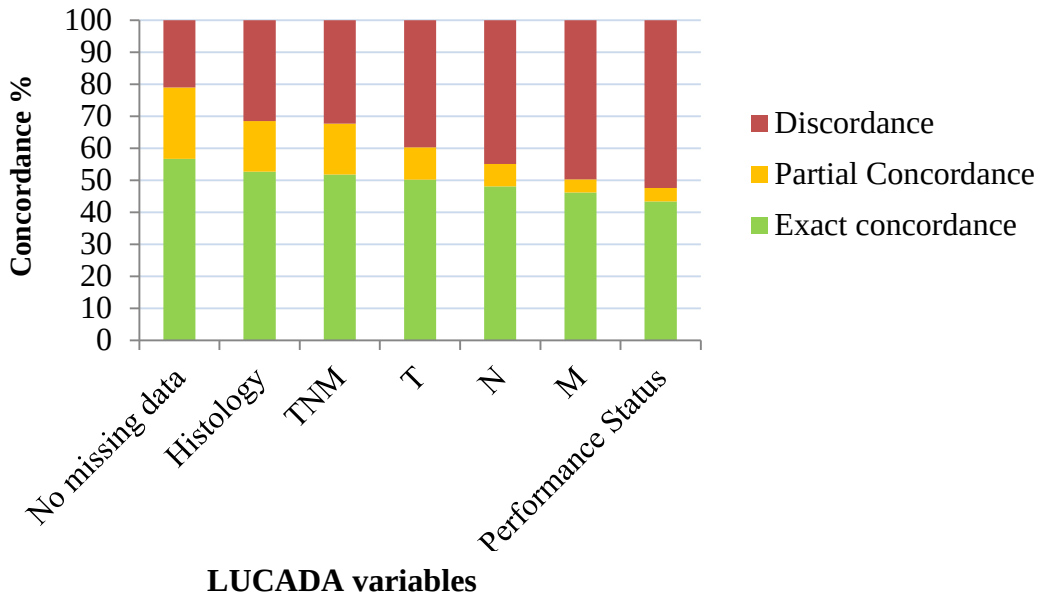

Figure 5.9: Concordance percentages with the cumulative increase of missing values for the "Histology", "TNM stage", "T stage", "N stage", "M stage" and "Performance Status" fields in the given order.

It is important to note that for the results reported in Figure 5.9, the removal of observed values plotted for each data field is permanent and, as a result, the level of missing data consistently increases with the removal of each data field. As can be seen in the figure, the exact and partial concordance percentages with no missing data are 57% and 79% as reported in the previous section. With the methodical removal of each additional data field from left to right, the concordance levels plummet since data required to infer rule eligibility becomes unavailable.

This is an inherent limitation of deterministic rule-based CDS tools. A possible way to deal with missing values within the boundaries of deterministic inference would be to explicitly model missing values with a "missing" state and create duplicates for each rule that can cater for all missing data combinations of different variables for a given rule. However, this would not be a desirable solution due to two reasons: 1) Firstly, by introducing rules that take into account incomplete observations, the knowledge engineer is in practice introducing haphazard rules that are no longer evidence-based and 2) for a rule criteria definition that employs four variables, introducing different versions of the rule that accounts for all possible combinations of incomplete observations results in 4! versions of the same rule. This would result in an undesirably high number of rules in the rule-base.

In their rule-based CDS prototype, MDTSuite, Austin et al. manage the problem of missing data by making use of the rule base to infer and highlight what data are required to be input by the clinicians in order to get meaningful results from the CDS tool [18]. While this approach is definitely useful in at least making clear the negative effects of missing data on recommendation quality to the end user, it also reveals the strong dependency of such deterministic systems on complete data observations. On the other hand, to our knowledge, the MATE [26] and OncoDoc2 [29] CDS tools for breast cancer MDT meetings , which we reviewed in Chapter 1, do not provide any means to manage missing information.

In this matter, we adopt a view point of Artificial Intelligence and CDS that is more in line with the sentiments of Korb and Nicholson: "…*artificial intelligence is the development of an agent which has some hope of overcoming problems on its own, rather than requiring engineers and domain experts to hold its hand constantly*" [115].

5.3. Discussion

Overall, transforming narrative guidelines into computer-interpretable formats remains a bottleneck in the development of rule-based deterministic CDS tools. More often than not,

rule definitions need to be operationalised in terms of the available knowledge base, implicit expert knowledge needs to be explicated, the undefined terms need to be clarified and inconsistent recommendations between different documents need to be managed carefully [213].

Our knowledge elicitation effort, based on our review of the four main national and international guideline rule documents in lung cancer care, revealed that ambiguous wording, which is still surprisingly prevalent in most guideline rule documents, is a major source of uncertainty that can cause substantial variations in the interpretations of identical rules between different expert panels.

Nevertheless, our system evaluation experiment results showed that the first version of LCA could produce meaningful decision support based on guideline rules. While our definitions of exact and partial concordances between the top system recommendations and the recorded treatments in the dataset, i.e. the 'silver standard', are not unbiased or definitive measures that validate our system recommendations, the results obtained at least provide sufficient evidence that the guideline rule-based decision support provided by LCA is capable of making sensible recommendations.

Although it is not methodologically ideal, the adoption of a silver standard for evaluating the system recommendations was necessary due to the time limitations of this research. A system evaluation based on gold standards, would require a minimum 5-year long pilot study that evaluates the effect of CDS on the 5-year survival statistics of the patient set.

One thing to note is that while refining our rule base, we intentionally refrained from trading off our rule-base integrity in exchange of improving concordance with the silver standard. Expanding the rule-base by adding local knowledge elicited through our meetings with the clinicians could result in over-fitting the rule-base to reflect current

clinical practice and defeat the purpose of having the decision support platform serving as an independent agent that upholds evidence-based practice guidelines.

Various studies have reported that local adaptation of national guidelines is an important strategy for achieving local ownership and relevance of guidelines in order to increase the likelihood of clinician adherence [225]–[228].

As an example, of immediate relevance, the guideline knowledge for the MDTSuite Colorectal MDT decision support system [18] consisted of 109 arguments extracted from national and international guideline documents and 134 arguments that were elicited from local clinical experts during interviews in order to augment the former group of arguments. The researchers report that this augmentation was necessary in order to cater for patient cases which were not covered by national and international guidelines.

While the necessity for localisation is understandable, it presents an impending danger of overfitting, in other words compromising on the generality of the rule base for the sake of boosting concordance with the practice habits of a local clinical team by adding overly complex, situation-specific rules. In our case, the availability of nation-wide data, as opposed to local hospital records used in [18], [26], also precluded the option of incorporating local knowledge in our rule base.

Chapter 6 - LUCADA Bayesian Network

To this point in the thesis, we have focused on decision support provided by an argumentation-based decision model, which depends on a deterministic rule base in order to produce patient-specific arguments for or against various decision options. In so doing, we represented and reasoned with our domain knowledge through an OWL-2 ontology that operates within the boundaries of Description Logics. In this chapter, we investigate an alternative approach, namely probabilistic inference, in representing and reasoning with our domain knowledge.

Knowledge engineering is concerned with integrating knowledge into computer systems in order to solve problems that normally require a high level of human expertise. In this context, rule-based and probabilistic approaches to clinical decision support (CDS) share a common goal: reasoning with uncertain information. They do, however, start from very different hypotheses. As briefly discussed in Section 2.3, proponents of rule-based deterministic systems claim that most real-life problems are often under-specified and ill-formulated and therefore must be dealt with qualitatively. In contrast, champions of probabilistic inference assert that probability calculus is the only mathematically correct way for representing degrees of belief and reasoning with uncertain information [115].

In Chapter 2, we introduced a Bayesian Network (BN) as a graphical model of a joint probability distribution over a set of random variables. We noted that BNs are suitable tools for probabilistic inference in the context of clinical decision support (CDS), since 1) their graphical component allows a visually more appealing and transparent probabilistic inference; 2) they can incorporate domain knowledge and prior beliefs during structure learning and parameterisation; 3) they facilitate parameter estimation due to their compact representation of the joint probability space; 4) they not only allow observational inference but also causal interventions; and 5) they can be used to query any given node in the

network and are therefore more versatile compared to other classification methods that are built based on specific outcome variables.

We begin this chapter by elaborating on the clinical need within the MDT meetings for probabilistic inference in order to answer clinically meaningful questions that cannot be answered by rule-based decision support alone. Following this, before we set out to design a domain specific causal BN for modelling the clinicians' decision making process, we review the literature on the use of BNs for decision support in cancer. Since we already covered the general concepts regarding the design stages of a BN earlier in Section 2.4, in this chapter we focus just on the practicalities of the main implementation stages such as variable selection, handling missing data, structure learning and parameterisation. Finally, we conclude the chapter by presenting our empirical results on the Bayesian scores and predictive performances of the BN structures learned by different algorithms. We select the structure, which provides the best trade-off between a plausible causal structure and acceptable Bayesian score, to integrate into Lung Cancer Assistant.

6.1. Background

6.1.1. The Clinical Need for Probabilistic Inference

In the previous chapter, we showed that the first version of LCA was capable of producing meaningful patient-specific recommendations based on clinical guideline rules. Our results were promising and in line with previous research [18], [26], [29] in that evidence-based decision support is important in reducing the unjustified variations in clinical practice. However, a purely rule-based decision support approach falls short in answering certain kinds of clinical questions that the MDT members face on a weekly basis when devising treatment plans for the patients presented in the meetings.

Through our interactions with our clinical collaborators and our attendance at the lung cancer MDT meetings in the Oxford University Hospitals, we formed a good

understanding of the fundamental questions that need to be answered in order to arrive at more informed treatment decisions by the clinicians. Confronted with a patient case in the MDT meeting, the answers to the questions of "What is the probability of survival for this patient?" and "How do different treatment decisions affect this probability?" generally drive the decision making process.

The answer to the first prognostic question is vital for the decision making of the clinicians in order to identify patients for whom more aggressive curative treatment modalities would be appropriate, as opposed to those with a poorer prognosis where the focus may be palliative care [3]. An accurate patient-specific prediction of survival can be used to stratify cancer patients into different risk groups and potentially aid in devising more personalised treatment plans [124]. Furthermore, predicted survival information can also be pivotal in managing patient and family expectations on treatment outcomes [229]. Meanwhile, the second question addresses the pragmatic goal of curative cancer care. Naturally, if the prognosis for the patient is poor, the end goal may be palliation and management of symptoms, rather than increasing the probability of survival.

6.1.2. Observational and Causal Inference

BNs allow representing and reasoning with domain variables through probabilistic inference. There is, however, no consensus on the causal interpretation of BNs, i.e. whether or not the edges in a DAG strictly indicate causal relationships between domain variables. When the structure of a BN is constructed manually, the DAG can be made to reflect a causal understanding of the domain by the expert. However, when the structure is learned automatically from data, it becomes harder to claim that the learned DAG strictly implies causality. As we have already discussed in Section 2.4.2, many structure learning algorithms can only score competing structures up to their Markov equivalences [142] and as a result it is impossible to learn a unique DAG for a BN based solely on data, which

makes the causality hypothesis questionable. Spirtes et al. term this issue as "statistical indistinguishability", discussing it at length in [138].

We take the view that the edges in a DAG should mainly be interpreted as probabilistic dependencies that also lend insight into causal relationships. This is in line with the viewpoint adopted by [115], [230], [231] that the probability distribution represented by a BN has an underlying causal structure. In addition, we contend that especially in the field of medicine, this underlying causal structure is more than hypothetical and manifests itself within guideline rules as well. Uncertainty permeates causality in medicine although it is not always made explicit. For example, there is a causal relationship between 'Age' and 'Survival', even though it may not be straightforward to pinpoint through which variables it operates in a dataset.

This debate on the causal interpretation of BNs also leads to a differentiation between observational and interventional inference. Going back to the two clinical questions we posed in the previous section, the first question on survival probability, which we can denote as "$P(Survival = Alive\,|Evidence) = ?$", can be answered via observational inference. Here, we are simply interested in the posterior distribution of our query variable: $Survival$, conditioned on the observed $Evidence$ for some nodes. The field of machine learning (ML) offers many different methodologies to answer such observational queries.

The second question, on the other hand, is of a different nature, whereby we inquire about the effect of a treatment plan option on the probability of survival for a given patient. This is denoted in probabilistic terms as $P(Survival = Alive\,|Evidence, T) = ?$", where T represents the treatment plan variable. Here, the reasoning is aimed at finding out the posterior distribution of $Survival$ conditioned on T, which is –unlike $Evidence$- unobserved at the time of asking the question. In other words, the question is hypothetical and cannot be answered simply by the values observed up to that point. In order to predict

what the survival probability is going to be, given different T states, we need to make a causal intervention, which allows us to ask "What if?" questions. This type of causal reasoning is highly important in CDS applications and is not possible with ML methodologies such as regression models that lack the capability of carrying out interventions [115], [231]. In Section 6.3.3 we will elaborate on the implications of causal intervention on the way the inference, i.e. belief updating, should be executed within a causal BN.

6.1.3. Bayesian Networks and Cancer

We have already reviewed the literature on the use of BNs for CDS in Section 2.4.1. In this section, we mainly review the literature on the utilisation of BNs in cancer care. Cruz and Wishart [232] report that the adoption of ML techniques for prognosis prediction and treatment selection is a relatively recent development. The existing literature on BNs and cancer mainly concerns applications to aid diagnosis, risk evaluation and survival prediction. Furthermore, among different cancer domains, there has been a concentration on applications in breast cancer [128], [233]–[236] as compared to BN applications in other types of cancer [124], [151], [229], [237]–[239].

In terms of relevant BN applications on survival prediction, in a study published in 2011, which aims to predict the 1-year life expectancy of 189 patients with skeletal metastases, Forsberg et al achieved good predictive performance with an area under the receiver operator characteristics curve (AUC) of 0.83 [229]. In a more recent study based on a substantially larger dataset containing 146,248 patient records, Stojadinovic et al. built a BN to carry out personalised survival prediction for colon cancer, reporting an AUC value of 0.85 [240]. Neither of these studies compared the suitability of different approaches in the causal discovery of the domain structure. In addition, both causal interventions and the feasibility of treatment recommendations by the BNs were out of the scope of both studies.

Focusing on lung cancer specific applications of BNs, in 2010 Jayasurya et al. designed a BN in order to predict survival in non-small cell lung cancer (NSCLC) patients treated with radiotherapy. They concluded that BN models achieve a higher predictive performance with missing data, compared to support vector machines and are therefore more suitable for the medical domain [124]. In a more technically oriented publication, Oh et al. proposed a BN structure learning algorithm that combined both physical and biological factors for predicting local failure in lung cancer [239]. However, both of these studies were based on datasets that contained limited numbers of patient records. - for one study in [239] only 18 patients- necessitating replication on much larger datasets.

In summary, the number of studies reporting the application of BNs to cancer is limited. Furthermore, apart from a handful of exceptions, most published results are from preliminary studies based on limited patient data. To our knowledge, no prior work, which takes into account histological, clinical and demographic information based on a national dataset of the size of LUCADA, exists in survival prediction or treatment recommendation in lung cancer.

6.2. Pre-processing the LUCADA dataset

Data pre-processing is a crucial step in any machine learning exercise since the reliability of a classifier highly depends on the quality of the data used [241]. In the context of machine learning, a classifier is a function that assigns a class label to instances described by a set of attributes [242]. In order to ensure a high level of data quality, real-world datasets need to be pre-processed before being used for training a classifier. We already described various data cleaning steps we had to take in dealing with noisy LUCADA data in Section 3.2.4. In this section we mainly focus on variable grouping and discretisation. Following this, we develop a strategy for dealing with missing data in the LUCADA dataset and finally discuss our rationale for the selection of variables to include in our BN.

It is important to note that the pre-processing choices reported in this section have been informed by various iterations of our BN design attempts on the LUCADA dataset over the course of the project.

In Chapter 3, we established that we are only interested in the non-administrative LUCADA variables that are clinically relevant to both treatment selection and treatment outcomes. Table 6.1 lists all such non-administrative fields within the LUCADA dataset.

Despite the flat and tabular way these variables are presented in Table 6.1, it is important to note that there is an underlying temporal structure that dictates an order and dependencies between them. To highlight this, we stratified all variables in the table with respect to their orders of appearance during the patient journey. The first tier -colour-coded in green- represents pre-treatment variables that are context-independent and necessary to make a treatment decision for every patient. The second tier -colour-coded in orange- represents individual treatment-specific variables that only become available based on the treatment decision made. And finally, the third tier –colour-coded in blue- contain variables related to the outcomes.

Code	Name	Cardinality	Completeness %
1	Sex	2	100
2	Age at time of diagnosis*	***5***	100
3	Staging Identifier*	2	99.8
4	Basis of Diagnosis	***5***	100
5	Referral Source	***5***	100
6	FEV1 Absolute Amount*	***4***	31
7	FEV1 Percentage*	***4***	25.3
8	Performance Status*	***5***	72.4
9	Number of Comorbidities*	***6***	15.6
10	Primary Diagnosis*	***6***	96.8
11	Tumour Laterality*	***5***	84.9
12	TNM Category*	9	75
13	Histology (SNOMED)*	***8***	66.9
14	Site-specific Staging Classification*	***4***	61.5
15	Cancer care plan intent	***4***	71.9
16	Treatment plan type	2	67.4

17	Suggested cancer treatment plan	***10***	85.6
18	Primary procedure(OPCS)	***12***	83
19	Excision margin	***5***	77.9
20	Chemotherapy treatment given	***6***	54.5
21	Radiotherapy treatment given	***6***	78.1
22	Radiotherapy anatomical treatment site	***9***	75.4
23	Was death related to treatment	2	17.6
24	Confirm PCI	2	22.2
25	Original treatment plan carried out	2	30.7
26	Original treatment plan failure reason	5	3.9
27	Reason first choice treatment not given	4	13.6
28	1-yr Survival	2	97.9
P(1, 2, 3, …., 28)		3.10E+18	65.28

Table 6.1: Expert elicited LUCADA variables to be included in building a domain representative Bayesian Network. The variables are grouped with respect to their temporal orders of appearance in the patient journey.

In order to make the probability space more tractable and avoid conceptual duplications, which are not allowed in causal BNs [115], we implemented a couple of variable groupings on the dataset fields introduced in Chapter 3. For instance, instead of having pre- and post-surgical versions of "TNM staging" and "Histology" variables, we chose to represent each as single entities containing the information available before treatment selection. In addition, due to the very low degree of completeness for the individual co-morbidity fields, we merged all available co-morbidity information under a single variable named "Number of comorbidities" that stores the information suggested by its name.

As already mentioned in Chapter 3, we also manually removed the records where 1) the patient was diagnosed with Mesothelioma; 2) the patient was given Brachytherapy (less than 100 patients); and 3) there did not exist any survival information. This reduced the number of observations available in the dataset from 126,987 to 117,426.

6.2.1. Handling Missing Data

As already noted in Section 5.2.3.3, missing data is a reality of clinical datasets [223], [224]. Consequentially, handling missing data is almost a prerequisite for a CDS application to be adopted in clinical practice [124]. As Table 6.1 reveals, LUCADA suffers

heavily from data incompleteness where missing data comprises approximately 35% of the dataset.

Given a dataset D, there are various ways to deal with missing data. Let us denote the observed part of D by D_{obs} and the missing part by D_{mis}, so that $D = \{D_{obs}, D_{mis}\}$. Depending on how the incompleteness of any particular variable is related to other variables, missing data is commonly modelled based on one of three different assumptions: 1) missing completely at random (MCAR), which indicates that the probability of an observation being missing is unrelated to the value of any other variable in D_{mis} or D_{obs}; 2) missing at random (MAR), which postulates that the incompleteness pattern in D_{mis} can depend on D_{obs} but not on D_{mis} itself; or 3) not missing at random (NMAR) which represents cases that do not fall under 1 or 2, i.e. ignorable missingness [243]. This third category necessitates modelling the incompleteness mechanism explicitly.

More formal definitions of these assumptions, given in terms of a probability model for missingness, can be found in [224]. There is a variety of methods to deal with MCAR and MAR missing data. These include 'case deletion', where the incomplete observations are omitted from the analysis, and 'simple imputation' techniques, where the missing values of a variable are filled in with the mean, median or modes of the observed values. In most cases, these methods are inadvisable since they introduce biases and distort the estimated variances and covariances (hence the conditional dependencies) in the dataset [244].

There are also more modern methods, such as Expectation Maximisation (EM) [245] and Multiple Imputation [224] that account for the uncertainty introduced by the missing observations in a more principled manner. However, their usage depends on the validity of the MAR assumption, without which they result in biased estimates [243]. In addition, both EM and MI are computationally intensive algorithms that may not be practically feasible for datasets with high rates of incompleteness and large number of variables.

MAR is an assumption whose validity cannot be tested directly from the data [246]. Graham advises that "*the best way to think of all missing data is as a continuum between MAR and NMAR*" and one has to decide whether the MAR violation in a given data set is significant enough to render the estimates of MI and EM invalid [243]. From our interactions with the NLCA staff, we were lead to believe that NMAR missingness was prominent in LUCADA. As a matter of fact, missingness patterns in clinical datasets are in general correlated with the clinical relevance of the missing values for a specific patient and therefore embody information [223], [247], which may often violate the MAR assumption.

The performance and speed of EM and MI deteriorate as the number of variables and the percentage of missingness increase. Specifically, more missing data means the EM algorithm converges (to a local maximum) more slowly [243] and MI requires generating a considerably high number of imputed datasets [248]. In addition, a large number of variables necessitate estimating a high number of parameters. The largely categorical nature of the data in LUCADA poses a practical constraint for MI specifically, since the categorical variables need to be 'dummy-coded' before carrying out MI [243]. According to this, a categorical variable with cardinality p is represented as p-1 dummy binary variables that represent the different levels of the categorical variable. Applying this to the 28 variables in Table 6.1 results in 113 (141-28) binary variables that need to be fed into MI.

These theoretical and practical concerns motivated us to treat missing data by explicitly modelling the "missingness" given the context. However, before carrying on we first wanted to verify that the pattern of missingness in LUCADA did indeed embody class discriminatory information. For this reason, we ran a set of experiments on the 28-variable subset of the dataset with 117,426 patient records. In the experiments, we chose 1-year survival as our binary outcome variable, $S= \{0, 1\}^{117,426}$, and separated the rest of the

dataset as our feature matrix. Following this, we prepared a logical missingness indicator matrix' I= $\{0, 1\}^{27*117,426}$, whose elements were 0 or 1 depending on whether the corresponding elements of the feature matrix were observed or missing. We input the indicator matrix, I, into the Naïve Bayes and Logistic Regression algorithms for predicting S. Naïve Bayes is a simple probabilistic classifier based on applying Bayes' theorem with strong independence assumptions between the predictive variables [249]. Logistic Regression is a form of regression analysis that determines the impact of independent predictive variables to classify the outcome of a categorical dependent variable [250]. It models the logit-transformed probability as a linear relationship with the predictor variables. The AUC values and predictive accuracy percentages achieved by 10-fold cross-validated experiments with each algorithm are given in Table 6.2.

	Average AUC	**Std. Dev. AUC**	**Average Accuracy %**	**Std. Dev. Accuracy**
Logistic Regression	0.77	0.029	75	0.34
Naive Bayes	0.76	0.027	73	0.32

Table 6.2: Area under the curve (AUC) and predictive accuracy performance results for the missing data indicator matrix (27*117,426) in predicting 1-year survival outcome.

These results reveal that the missing data pattern is actually informative in predicting 1-year survival in the LUCADA dataset. Within the Logistic regression results, we further analysed the coefficients given to different binary indicator variables in the dataset, which are plotted in Figure 6.1. For the sake of visual clarity, within the plot we replaced the variable names with the corresponding codes listed in Table 6.1.

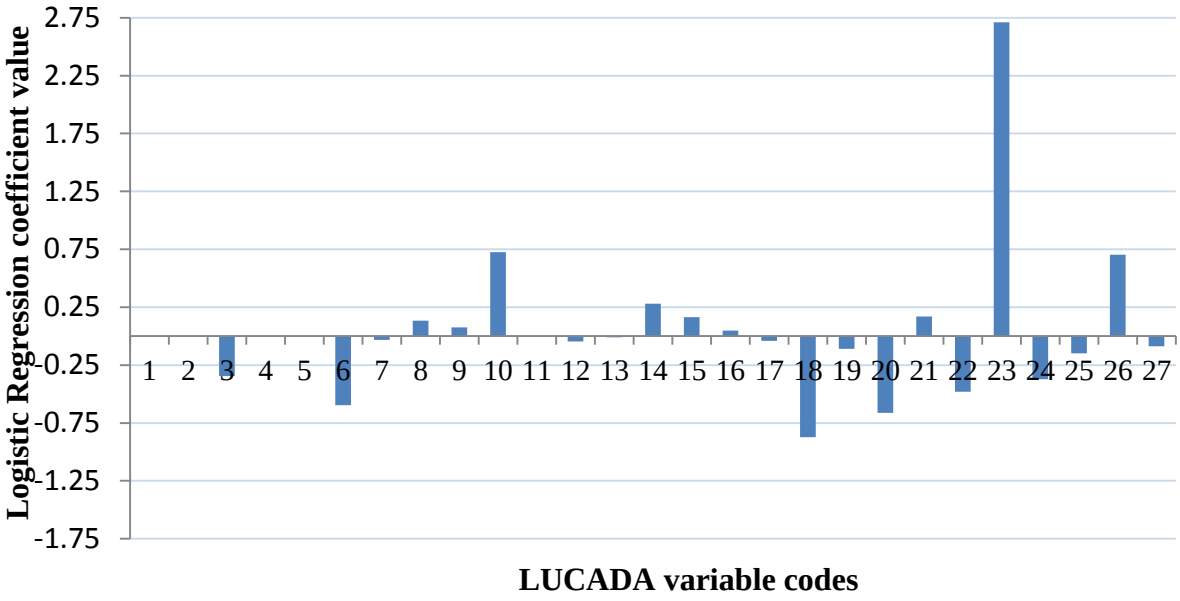

Figure 6.1: The logistic regression coefficients for the LUCADA variables, whose absence have been modelled in the missing data indicator matrix.

As can be seen in Figure 6.1, the 5 variables whose data absence has the highest impact on mortality prediction are: "Was Death Related to Treatment" (23), "Primary Surgery Type" (18), "Original Plan Failure Reason" (26), "Primary Diagnosis" (10) and "Chemotherapy treatment given" (20). Two out of these five variables are indeed outcome-related. Contextually, the information of the "Was Death Related to Treatment" and "Original Plan Failure Reason" fields being filled in is a strong indicator of mortality. In general, all variables whose absences have strong influences on mortality are highly context-dependent and contain information that is actually not available at the time of treatment selection. We will discuss this issue in more detail during variable selection in Section 6.2.3.

In order to re-evaluate the predictive performance of the missing data indicator matrix after the exclusion of the outcome related variables that are highly indicative of mortality, we repeated the experiments, this time including only the pre-treatment and treatment variables in the logical indicator matrix. The results for this second set of experiments are given in Table 6.3.

	Average AUC	**Std. Dev. AUC**	**Average Accuracy**	**Std. Dev. Accuracy**
Logistic Regression	0.72	0.024	72	0.37

Naive Bayes	0.69	0.021	71	0.36

Table 6.3: Area under the curve (AUC) and predictive accuracy performance results for the reduced missing data indicator matrix (23 × 117,426) in predicting 1-year survival outcome.

As can be seen in Table 6.3, despite a slight decrease caused by the removal of the outcome-related variables, the predictive performance of the indicator matrix remains fairly high. These results indicate that the incompleteness pattern in the LUCADA dataset holds a significant amount of class discriminatory information for the 1-year survival outcome variable.

For this reason, we opted to model missing data explicitly in our analyses. In doing so, we used PostgreSQL queries to replace the null observations in the database with an explicit "Unknown/Missing" state. However, we were aware that some variables were just missing due to contextual irrelevance, given the recorded treatment plan type for a patient. We already gave some examples of these in section 3.2.2 when we introduced treatment-related variables in the LUCADA dataset. As an example, when a patient is treated with chemotherapy, we would expect the "Chemotherapy Treatment Given" variable to be observed. However, other variables which do not concern chemotherapy, such as "Primary Procedure" and "Radiotherapy Treatment Given", become irrelevant for that patient record. In order to distinguish this type of incompleteness from data which were missing without any obvious pattern, we treated such contextual incompleteness in the dataset with a "Not Applicable" state. The reader may notice that the completeness levels for some treatment-related variables in Table 6.1 are significantly higher than the figures previously reported in Table 3.4 in section 3.2.2. This is due to the explained introduction of the "Not Applicable" state for some variables during data pre-processing for our analyses.

Another interesting observation was that according to the LUCADA data model, some variables such as "Performance Status" and "Treatment Plan Type" could take on "Unknown" status. Since we chose to model missing values with an explicit "Unknown/Missing" state and having two separate states that represent lack of information

would not be ideal, for such variables we merged the "Unknown" status present in the database with our manually added "Unknown/Missing" state.

6.2.2. Discretisation

As already discussed in Chapter 3, apart from "*Age*", "*FEV1 Percentage*" and "*FEV1 Absolute amount*", all the variables in the LUCADA data model are categorical. While it is possible to build BNs with continuous variables, the majority of clinical applications comprise just categorical variables [114]. As a matter of fact, in many BN applications in the literature, variables are discretised into two or three bins in order to keep inference tractable [239].

There are various binning techniques for automatic discretisation of continuous variables based on different information criteria [251]–[253]. However, in our use case, the availability of certain cut-off values within the guideline documents and our interactions with our clinical collaborators allowed us to perform manual discretisation based on clinically meaningful intervals. These expert elicited intervals are listed in Table 6.4.

	Age	**FEV1 %**	**FEV1 Absolute Amount**
1	< 50	< 30	< 1.0
2	50-60	30-40	1 - 1.5
3	60-70	40-80	1.5 - 2.0
4	70-80	> 80	> 2.0
5	> 80		

Table 6.4: Expert-elicited bin intervals used in the manual discretisation of the "*Age*", "*FEV1 Percentage*" and "*FEV1 Absolute amount*" variables.

In this table, the "FEV1 Percentage" binning intervals are derived from a mixture of NICE Chronic Obstructive Pulmonary Disease Guidelines [254] and our expert panel's suggestions. The intervals for "FEV1 Absolute amount" were informed by the cut-off values implemented by the MDT team in determining suitability for pneumonectomy, lobectomy and radiotherapy treatments.

6.2.3. Variable Selection

Having completed the necessary pre-processing steps, we performed variable selection to choose those variables on which we would base our probabilistic domain representation. In general, variable selection strives to eliminate all but the most relevant features with the aim of reducing the dimensionality of a given probability space and yield a more compact and easily interpretable representation of the target domain [255]. Striking the right balance between preserving class discriminatory information and minimising domain complexity is a crucial step in any machine learning exercise.

Through our guideline rule elicitation efforts in Chapter 5, we already uncovered a list of the most commonly encountered variables in lung cancer care guideline rule definitions, listed in Table 5.2. It is highly likely that this list constitutes the primary set of parameters that inform treatment selection and outcome in the lung cancer domain and are marked with asterisks in Table 6.1.

As the reader may notice, all guideline variables in Table 5.2, with the exception of 'Excision Margin', fall in the pre-treatment category. This is due to the fact that the treatment selection related guideline rules we elicited are aimed at informing treatment decisions in the MDT meetings and to that end they mainly focus on variables that are likely to be available at the time a new patient is presented for a treatment decision in the MDT. Since our CDS prototype is designed with the same motivation, we opted to base our domain representation on the same 11 pre-treatment variables that we used in the previous chapter for rule-based decision support. With the addition of our two designated query variables: "Suggested Cancer Treatment Plan" and "1-year survival", the number of variables we selected amounted to 13.

Theoretically, we could add all 28 variables to acquire a more comprehensive domain representation. However, it is important to bear in mind that the addition of treatment-related and outcome related variables would increase the complexity of the joint

probability space at practically no gain, since in an ordinary use case no variables apart from those colour-coded in green in Table 6.1 are available to the clinicians at the time of selecting a treatment. As a matter of fact, in such predictive models, the exclusion of variables that occur after the predefined intervention time point –for our purposes, treatment selection- is common practice [256]. Our choice of a consistent set of variables with those used in the previous chapter was also convenient in order to facilitate the comparison of argumentation-based and probabilistic treatment recommendations provided by LCA.

In summary, our variable selection was primarily motivated by contextual relevance and availability of variables at the time of providing decision support. Nevertheless, we wanted to evaluate the performance of different variable subsets in predicting one-year survival in order to improve our understanding of the domain. For this purpose, we ran a set of experiments that included "Suggested Cancer Treatment Plan" and "1-year survival" variables in addition to these three variable subsets: 1) Guideline variables (marked with asterisks in Table 6.1 and colour-coded as purple in figures 6.2 and 6.3); 2) All Pre-treatment variables (colour-coded in green in Table 6.1); and 3) All pre-treatment and treatment related variables (colour-coded in green and orange in Table 6.1). As such, the subsumption relations between subsets 1 to 3 can be given as: Set 1 $\subset$ Set 2 $\subset$ Set 3.

Since the survival prediction only requires observational inference, we chose to run our classification experiments using the Naïve Bayes and Logistic Regression algorithms available in MatLab R2011a and the C4.5 decision tree implementation in the WEKA 3 machine learning tool [257]. Figures 6.2 and 6.3 depict AUC and predictive accuracy results obtained as a result of our 10-fold stratified cross validation experiments with each algorithm.

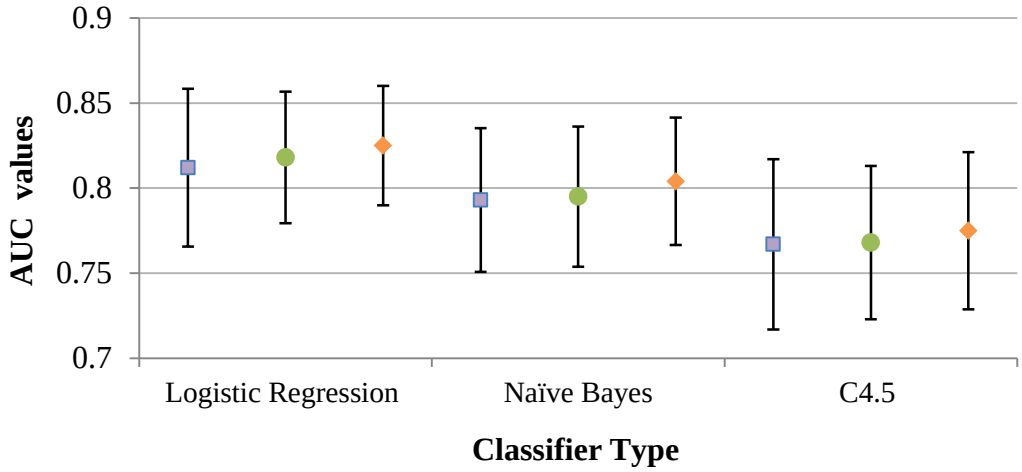

Figure 6.2: The average 10-fold cross-validation AUC results with their standard deviations. The results for subsets 1, 2 and 3 are marked with purple squares, green circles and orange diamonds respectively.

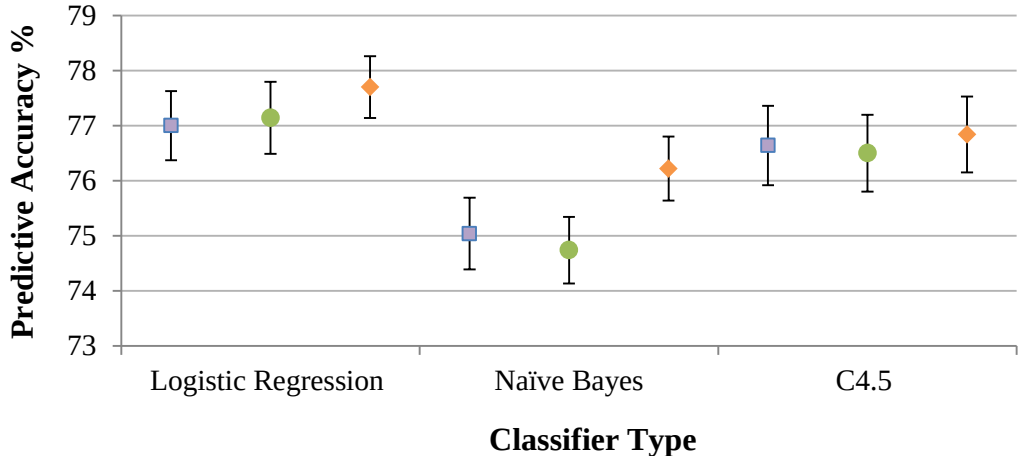

Figure 6.3: The average 10-fold cross-validation predictive accuracy results with their standard deviations. The results for subsets 1, 2 and 3 are marked with purple squares, green circles and orange diamonds respectively.

As is clear from both figures, logistic regression provides marginally higher AUC and predictive accuracy results compared to the Naïve Bayes and C4.5 algorithms. Focusing on the intra-classifier variations of AUC values and predictive accuracies obtained with the three variable subsets, we can see that the inclusion of additional variables on top of subset 1 (marked with purple squares) does not result in significant improvements in predictive performance. This is an encouraging finding which suggests that basing our domain representation on the 13-guideline-variable subset does not deteriorate the predictive performances of the classifiers greatly.

6.3. Experimental Methods

Having performed the necessary pre-processing steps, we design our BN based on the 13-variable subset, which is presented separately in Table 6.4, using the same colour-coding as in Table 6.1. Building CDS systems based on exemplar data like LUCADA is an emerging field of research, which is likely to become more prevalent as the availability of machine-interpretable patient data increases through wider adoption of electronic health records.

Code	Name	Cardinality
1	Age at time of diagnosis*	*5*
2	Staging Identifier*	2
3	FEV1 Absolute Amount*	*4*
4	FEV1 Percentage*	*4*
5	Performance Status*	*5*
6	Number of Comorbidities*	*6*
7	Primary Diagnosis*	*6*
8	Tumour Laterality*	*5*
9	TNM Category*	9
10	Histology (SNOMED)*	*8*
11	Site-specific Staging Classification*	*4*
12	Suggested cancer treatment plan	*10*
13	1-yr Survival	2
Total size of the probability space		**8.29E+08**

Table 6.4: The 13-guideline variables to be included in the design of the LUCADA Bayesian Network

Driven by the clinical need presented in the beginning of the chapter, our primary focus in this section is to utilise various well-established methodologies for building a domain-specific BN that can be used to answer observational and interventional queries on different variables in order to aid the clinicians in their decision making. While classification performance is important in terms of the reliability of the predictive answers, our choice of BNs as the domain representation methodology is mainly motivated by the need to create a generative and causal model that mimics the decision making of the clinicians during treatment selection.

In the remainder of this section, we will introduce the main parameter learning, structure learning and inference algorithms that we used in our experiments. As with the vast majority of ML studies –excluding time series analyses- we will also make the general assumption that the observations in the dataset are independently and identically distributed (iid). All experiments in this chapter were carried out by partitioning the selected 117,426-patient-strong subset of the LUCADA dataset into 10 equally-sized parts with approximately equal prior outcome probabilities, where probability of 1-year survival is 0.33. For each BN experiment, structure and parameter learning have been performed on the union of 9 partitions and tested on the remaining one. By iterating this process over all ten partitions, we ensured inclusion of all patient records in the experiments. The performances of all causal BNs and other predictive models were evaluated based on the AUC values and predictive accuracy percentages of stratified ten-fold cross-validations.

6.3.1. Baseline Benchmarking Algorithms

While the main focus of this chapter is on BNs, in the results section we also report classification performances obtained by the Naïve Bayes (NB), Logistic Regression, and C4.5 decision tree algorithms in order to provide references based on these algorithms that are commonly used for classifying clinical datasets.

We made use of the NB algorithm in MatLab R2011a and Logistic Regression and C4.5 decision tree algorithms available within WEKA 3 [257]. NB has been adopted as the baseline performance metric in many classification studies. Despite its simplicity, it has been reported to yield comparable results to more sophisticated ML techniques, especially in the presence of large datasets [249], [258]. Logistic regression is commonly used in clinical cohort studies and trials [259]. The specific implementation of Logistic Regression in WEKA 3 is based on using ‘ridge estimators’ for improving coefficient estimates [260]. C4.5 is a commonly used algorithm for building decision trees, which are deemed to be

more suitable for domains with discrete variables like ours [261], [262]. The specific implementation of the C4.5 algorithm that we used in WEKA 3 is named J48.

Among these algorithms, NB is similar to BNs in being generative. In ML, models that explicitly or implicitly model the distribution of inputs as well as outputs are termed 'generative' because by sampling from them, it is possible to create synthetic data points in the input space [263], [264]. On the other hand, logistic regression and decision trees are termed as discriminative classifiers that model the posterior probabilities directly. While discriminative models are less computationally intensive, as discussed in Section 6.1.2, such models are not fit for carrying out causal interventions on input variables.

6.3.2. Bayesian Network Design

We already introduced the general design stages of a BN in section 2.4.2. In this section, we will focus on introducing the specific methodologies and algorithms that we made use in order to learn the structure (DAG) and parameters of the LUCADA BN in our experiments. Before we begin our discussion on parameterisation and structure learning, it is worthwhile to establish a notation to be used for brevity. Let X_i be a discrete variable in a BN, and $Pa_{(Xi)}$ represent the parent nodes of X_i. Denote the cardinality of X_i by r_i and the cardinality of all state combinations for $Pa_{(Xi)}$ by q_i. According to this, the conditional probability table (CPT) for X_i will contain $r_i \times q_i$ elements.

Since the LUCADA dataset is made up of discrete variables, we make use of state counts for parameterising and scoring our models. this will be explained in more detail in the next section. The experimental set-up via which we learn the structure and parameters and report predictive performance metrics with each algorithm is summarised in Figure 6.4.

Experimental Setup:

Input: data (**D**)

for (xv=1 to 10)

D (xv) → **D training** (xv) + **D test** (xv)

DAG (xv) ← learn structure (**D training** (xv), **p**)

BN (xv) ← parameterise (**DAG** (xv), **D training** (xv))

[**Perf. metrics** (xv)] ← predict (**D test (xv)**, **BN** (xv))

end for

$\mathbf{DAG}_{\mathbf{final}}$ ← MWST (**DAG**)

Bayesian Score = score Dag (**D**, $\mathbf{DAG}_{\mathbf{final}}$)

Output: { DAGfinal , Perf. metrics, Bayesian Score}

Figure 6.4: The experimental setup via which we determined the DAGs and performance metrics for different structure learning algorithms.

As can be seen in Figure 6.4, for each fold of cross-validation, we separate the dataset **D** (xv) into training and test sets. We use the training set for learning the structure (DAG) and parameters of the BN and the test set to evaluate the predictive performance of the learned structure. It is important to note that in this experimental set up, we store **DAG** (xv) for each fold in the form of a logical adjacency matrix. At the end of the cross validation, we input the **DAG** array, which consists of all structures learned during the 10-fold cross validation, into a directed maximum width spanning tree (MWST) algorithm in order to acquire the resulting $\mathbf{DAG}_{\mathbf{final}}$. We then make use of the Bayesian Score metric as given in the next section in order to calculate P (**D**, $\mathbf{DAG}_{\mathbf{final}}$).

We implemented a MWST algorithm in MatLab by simply reversing the primary steps in Kruskal's minimum spanning tree algorithm [265]. As can be seen in Figure 6.5, we start by summing up all adjacency matrices in **DAG** in order to form the weighted directed graph, **WDG**. Each cell in **WDG** –denoted as a_{ij}- therefore represents the cumulative occurrences of the edges from the i^{th} to the j^{th} variable.

Directed Maximum Width Spanning Tree Algorithm:

Input: DAG (1 … xv)

$$\mathbf{WDG} = \begin{bmatrix} 0 & a_{12} & a_{1N} \\ a_{21} & \cdots & a_{2N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & 0 \end{bmatrix} \leftarrow \text{sum } (\mathbf{DAG}\ (1 \ldots xv)\ )$$

sorted edges ← sort (**WDG**)

$\mathbf{DAG_{final}} = \text{zeros}(N,N)$

for (i=1 to length(**sorted edges**))

    DAG' = $\mathbf{DAG_{final}}$ + **sorted edges** (i)

    if graphisDAG(**DAG'**)

        $\mathbf{DAG_{final}} \leftarrow \mathbf{DAG_{final}}$ + **sorted edges** (i)

    end if

end for

Output: DAG$_{final}$

Figure 6.5: The MWST algorithm we implemented to calculate the final DAG for all structure learning algorithms.

We then sort all edges in **WDG** in descending order and start by adding the edge with the largest weight to $\mathbf{DAG_{final}}$. We repeat this process for all remaining edges in descending order as long as the edge addition does not form a cycle. For checking whether a proposed edge addition causes a cycle, we make use of the "*graphisdag()*" function available in the Bioinformatics Toolbox for MatLab R2011a.

6.3.2.2. Structure Learning

In section 2.4.2.3, we noted that BNs can be designed either by eliciting expert knowledge or by various algorithms that make use of available data for uncovering the causal structure. In this section, we will briefly cover the main structure learning algorithms that we use in learning a causal structure based on the LUCADA dataset. The automatic learning algorithms introduced here were implemented either in the MatLab BNT toolbox [118] or the WEKA 3 [266] machine learning software. For hybrid structure learning, we used the CaMML software [267] developed at Monash University, Australia. All

algorithms we used assumed the BN variables to be discrete and the dataset to be fully observed.

6.3.2.2.1. Constraint-based Algorithm

The constraint based methods are focused on recovering a causal structure based on conditional independencies in the data. In our experiments we made use of an improved version of Inferred Causation (IC) algorithm as described in [268] and implemented by Bouckaert in WEKA 3 [266]. Starting with a complete undirected graph, the IC algorithm first tries to find conditional independencies, i.e. P (X_i,X_j|S), in the data through statistical independence tests in order to recover an undirected skeleton. The DAG is then returned by directing all edges in the skeleton following an order that enforces structural consistency.

6.3.2.2.2. Score-based search algorithms

All score-based search algorithms make use of decomposable score metrics that allow the total score for a DAG to be calculated as the sum (or product) of the individual node scores in the network. The Bayesian approach to structure learning amounts to searching for the DAG (G) with the highest relative posterior probability given a dataset (D) [269]. Given two competing structures, G1 and G2, the basic principle is given as in (6.1).

$$\frac{P(G1|D)}{P(G2|D)} = \frac{\frac{P(D|G1)\times P(G1)}{P(D)}}{\frac{P(D|G2)\times P(G2)}{P(D)}} = \frac{P(D|G1)\times P(G1)}{P(D|G2)\times P(G2)} \quad \textbf{(6.1)}$$

Therefore, finding the DAG that maximises P(G|D) is equal to finding the DAG that maximises P(G,D) [143]. The ratio $\frac{P(D|G1)}{P(D|G2)}$ in (6.1) is referred to as the Bayes factor, which is the Bayesian equivalent of the likelihood ratio test [136]. The usage of the Bayes factor has the effect of penalising models with too many parameters [118], [270]. Based on (6.1), various Bayesian scores have been derived with different assumptions by [143], [271],

[272]. The general Bayesian score formula for a completely observed and discrete dataset, containing N observations, is given in (6.2).

$$P(D,G) = P(G) \times \prod_{i=1}^{n} \prod_{j=1}^{qi} \left(\frac{\Gamma(N'_{ij})}{\Gamma(N_{ij} + N'_{ij})} \times \prod_{k=1}^{ri} \frac{\Gamma(N_{ijk} + N'_{ijk})}{\Gamma(N'_{ijk})} \right) \qquad \textbf{(6.2)}$$

In (6.2), Γ is the Gamma function and P(G) is the prior on the network structures. N_{ij} represents the number of observations in D for which $Pa_{(Xi)}$ takes its j^{th} value and N_{ijk} specifies the number of observations of a particular state of X_i in combination with the j^{th} value of $Pa_{(Xi)}$. On the other hand, N'_{ij} and N'_{ijk} represent the choices of priors on counts restricted by $N'_{ij} = \sum_{k=1}^{ri} N'_{ijk}$.

With $N'_{ijk} = 1$ (thus $N'_{ij} = ri$), we obtain the K2 score [143]; whereas with $N'_{ijk} = \frac{1}{ri \times qi}$, we obtain the Bayesian Dirichlet Equivalent (BDeu) score [272] [269] . For a detailed empirical comparison of the BDeu and K2 scores, and in particular the effect of likelihood equivalence in structure learning, the reader is referred to [273].The specific score-based search algorithms that we employed based on the Bayesian score were:

Tree Augmented Naïve Bayes

The Naïve Bayes algorithm assumes that all predictive variables are conditionally independent given the outcome variable. Friedman and Geiger introduced the tree-augmented Naïve Bayes (TAN) algorithm as a relaxation of this strong independence assumption between the predictor variables by using a tree structure imposed on the naïve Bayesian structure [242]. In our experiments, we used the WEKA 3 implementation of the TAN algorithm, which uses the maximum weight spanning tree algorithm by Chow and Liu [274] to form the tree imposed on the NB structure. This aims to find an optimum set of n-1 first order dependence relationships among the n variables using mutual information.

K2

As mentioned in Chapter 2, exhaustive search over the space of the DAG structures is computationally not feasible. K2 is the earliest score-based search algorithm [143] that addressed the issue of learning a causal structure. In order to reduce the search space, the algorithm assumes a prior knowledge of the topological ordering on the nodes. It then operates on a simple greedy hill climbing algorithm to test parent insertions according to the manually input order. In our experiments, we made use of the BNT toolbox implementation of the K2 algorithm, where we input the variable ordering as given in Table 6.4 and capped the maximum number of parents that an individual can have to six.

MC^3

Markov Chain Monte Carlo (MCMC) methods are dominant in stochastic search and approximate inference in the Bayesian statistics community [118]. The MCMC Model Composition (MC^3) algorithm proposed by Madigan and York [275] is used to draw samples from the posterior distribution of $\mathrm{P(G|D)}$. To construct the Markov chain, they define a neighbourhood for each graph structure G, i.e. $nbd(G)$, which consists of all DAGs that differ from G by one edge modification subject to the acyclicity constraint. The MC^3 algorithm is summarised as in Figure 6.6.

Algorithm MC³:

Input: G_0, B, N

for (i=0 to B+N)

Sample $G' \sim$ Q (G'|G)

Sample u $\sim U_{|0,1|}$

Compute $R = \dfrac{P(G') \times P(D|G') \times Q(G_i|G')}{P(G_i) \times P(D|G_i) \times Q(G'|G_i)}$

if (u < min(1, R))

$G_{i+1} = G'$

else

$G_{i+1} = G_i$

end if

end for

Output: { G_{B+1}, ..., G_{B+N}}

Figure 6.6: The pseudo algorithm MC3 algorithm implemented in the MatLab BNT Toolbox by Murphy [118]

In this definition, B is the burn-in period and N is the size of sampled DAGs the algorithm outputs in the end. The burn-in period is necessary to ensure that the Markov chain has reached its stationary distribution before starting to take samples. In the calculation of R, the conditional probabilities, $P(D|G')$ and $P(D|G_i)$, are acquired per the Bayesian score in (6.2). On the other hand, the transition matrix $Q(G'|G_i)$ is defined by setting $Q(G'|G_i) = 0$ for all $G' \notin nbd(G_i)$ and $Q(G'|G_i) = 1/|nbd(G_i)|$ for all $G' \in nbd(G_i)$. Finally, the algorithm outputs N sampled graphs from the approximate stationary distribution reached after discarding B samples.

In our experiments, we used the MC^3 algorithm as implemented by Murphy in the MatLab BNT toolbox [118]. For each experimental run, we initiated the algorithm with a fully disconnected DAG, and discarded the first 2000 samples before taking samples. We set our sample size per experiment to 1000. At the end of each experimental run, we obtained the

DAG structure by running the MWST algorithm, explained in Figure 6.5, on the sampled structures.

Simulated Annealing

Simulated annealing is an optimisation 'heuristic' that is employed in order to find a global optimum in large search spaces that are likely to be multimodal, which renders greedy search algorithms prone to getting stuck in local optima [276]. It is an adaptation of the Metropolis-Hastings algorithm, similar to MC^3 explained in the previous section. The name derives from annealing in metallurgy, which allows controlled cooling of a material to reduce its structural defects.

According to the algorithm, a starting temperature T determines the probability that a step in the 'wrong' direction in the search space is accepted. As such, T acts as the limiting factor in determining the size of the search space that can be visited during the stochastic iterations. In addition, a delta factor defines the rate by which the temperature, i.e. the acceptance probability of transitioning into a less likely state, is decreased with each iteration. There are various implementations of simulated annealing to BN learning in the literature. In our experiments, we made use of the algorithm implemented by Bouckaert in WEKA 3 [266]. We used the default starting temperature of 10 and delta value of 0.999, keeping the number of iterations at 10,000 for each experiment.

6.3.2.2.3. Hybrid Structure Learning

Among the score-based algorithms that we used in this chapter, CaMML is the only one that makes use of an information-theoretic metric, namely Minimum Message Length (MML). In contrast with the Bayesian scores explained in the previous section, MML is a non-Bayesian function that evaluates the goodness of fit of networks to data [277]. MML[141] was inspired by Solomonoff's early work on information-theoretic induction[278]. The main principle behind it is to find a trade-off between model

simplicity and fit to the data by minimising the length of a joint description of the model and the data assuming the model is correct[115].

CaMML carries out a two-stage search that is initiated by simulated annealing and continued by a Metropolis (MCMC) search over the probabilistic model space; a detailed description of the algorithm can be found in [115]. It has previously been used by Flores et al [122] and Twardy et al [129] to learn clinical causal structures in the domain of cardiovascular disease.

The appeal of CaMML for our purposes originates from its ability, as the most comprehensive hybrid structure learning algorithm, to incorporate different types of expert knowledge into the learning process. In our CaMML experiments we incorporated expert knowledge in the form of: temporal orders (A happens before B, denoted as $A \prec B$), direct relations (A and B are related, denoted as $A - B$) and direct causal connections (A directly influences B, denoted as $A \rightarrow B$).

We made use of the Java implementation of the algorithm developed at Monash University for running structure learning experiments with CaMML. For this purpose, we had to manually input the 10-fold stratified data sets into the program and transfer the output structures for each fold into MatLab in order to calculate the performance metrics and acquire the global DAG through the MWST algorithm explained in Figure 6.5.

6.3.2.1. Parameter Learning

In all BN experiments, we represented the joint probability distributions using CPTs as explained in section 2.4.2.3. For parameterising the CPTs, we used Maximum a Posteriori (MAP) estimations on the local multinomial distributions of variables. Using the notation introduced at the beginning of Section 6.3.2, the MLE of the probabilistic parameters, $\theta_{ik} = P(X_{ik}|Pa_{(X_i)})$, for a discrete variable X_i can be determined by maximising equation 6.3.

$$P\left(D|\theta_i\right)=\frac{N!}{\prod_{k=1}^{ri}N_{ik}!}\times\prod_{k=1}^{ri}\theta_{ik}{}^{N_{ik}} \quad \textbf{(6.3)}$$

Due to the probabilistic definition (6.4), maximising (6.3) becomes a constrained optimisation problem. Using Lagrange multipliers, the MLE for X_i can be reduced to give (6.5) as explained in [279].

$$\sum_{k=1}^{ri}\theta_k=1 \quad \textbf{(6.4)}$$

$$\theta_{ik}=\frac{N_{ik}}{\mathrm{N_i}} \quad \textbf{(6.5)}$$

Generalising (6.5) to take into account the state combinations of the parents of X_i, i.e. $\frac{N_{ijk}}{\mathrm{N_{ij}}}$, and incorporating our prior beliefs into the MLE estimation in the form of pseudo counts, i.e. N'_{ijk} and $\mathrm{N'_{ij}}$, we acquire (6.6) that gives MAP estimations of the parameters:

$$P(\mathrm{X_i=k\,|Pa_{(Xi)}}=j)=\frac{N_{ijk}+N'_{ijk}}{\mathrm{N_{ij}+N'_{ij}}} \quad \textbf{(6.6)}$$

As already discussed in the case of binomial distribution in Section 2.4.2.3, here $\mathrm{N'_{ij}}$ represents the equivalent sample size, where each constituent N'_{ijk} stands for the hyperparameters of a Dirichlet distribution, through which the knowledge engineer can pull the MLE towards their prior beliefs. If $\mathrm{N'_{ij}}$ and $\mathrm{N'_{ij}}$ are set to 0, we simply obtain the MLE of the given CPT element for X_i.

Overall, this provides a simple counting solution to the problem of parameterising multinomial networks and is therefore the most commonly used parameterisation solution in the vast majority of BN applications [115]. For all BN experiments presented in this chapter, we performed parameter learning by assuming uniform Dirichlet prior distributions over all discrete variables and by computing MLEs in order to learn the

conditional probability tables. This means that the experimental results are comparing the variation of the structure learning algorithms by using the same parameterisation technique.

6.3.2.3. Inference

As emphasised earlier, one of our reasons to represent our domain as a BN is the versatility of probabilistic inference provided by BNs, whereby entering evidence on any variable in the network results in updating the posterior distributions of the rest of the variables. These probability updates, i.e. belief updates, can be visualised on top of the graph structures, providing a certain level of transparency during inference and differentiates BN inference from "black-box" ML processes [115].

In our experiments, we made use of the Junction Tree algorithm [280] as separately implemented by Murphy [118] in MatLab BNT toolbox and Bouckaert [266] in WEKA 3. This algorithm consists of 'moralising' and 'triangulating' a DAG structure to create a junction tree structure over which a message passing algorithm is run for belief updating. As mentioned in Section 6.1.2, the usage of such a message passing algorithm has certain implications in belief updating with causal interventions [115]. This can be explained better with a context-specific example as given in Figure 6.7.

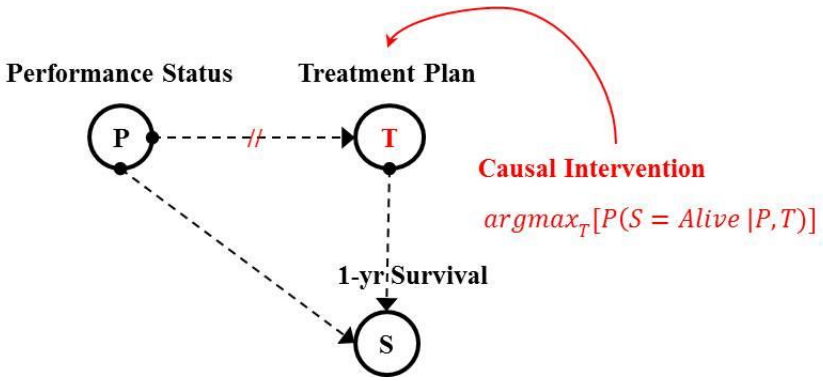

Figure 6.7: Schematic explanation of a causal intervention query on a BN.

The message passing algorithm operates through forward and backward propagation of observed evidence in the graph. As a result, when we are manually intervening on T as shown in Figure 6.7, Pearl suggests that all edges from the parents of T to T need to be removed in order to eliminate the indirect causal path connecting T to S through P [231]. Or putting it more simply, the direct intervention on T should render the effects of all parents of T on T ineffective.

6.4. Causal Structure Learning Results

In order to discover a structure that encapsulates the causal domain knowledge of the clinicians, while achieving a high Bayesian score and predictive performance, we tried various causal discovery approaches based on the structure learning algorithms explained in the previous section. In addition, we worked together with our clinical collaborators in building a manual structure and also in incorporating their expert knowledge into the learning process in different ways. In this section, we present both the Bayesian scores and the predictive performances achieved by the BNs whose DAGs were learned with different approaches. We also present the performance of the aforementioned baseline benchmark algorithms on predicting the "1-yr Survival" outcome to serve as a reference. Consequentially, the analysis of the BN results are carried out in two dimensions, focusing on both a measure of fit of the causal structure to data, i.e. Bayesian score, and the predictive performances of the proposed networks.

The expert elicited causal structure of the domain is given in Figure 6.8. This structure was built by guiding the members of our expert panel to connect the 13 domain variables based on a notion of causality, more specifically, asking them to point out the direct influences each variable has on others. As can be seen in Figure 6.8, there is limited interaction between the pre-treatment variables (1-11) and the edges often point from the pre-treatment variables to the treatment selection (12) and treatment outcome (13) variables.

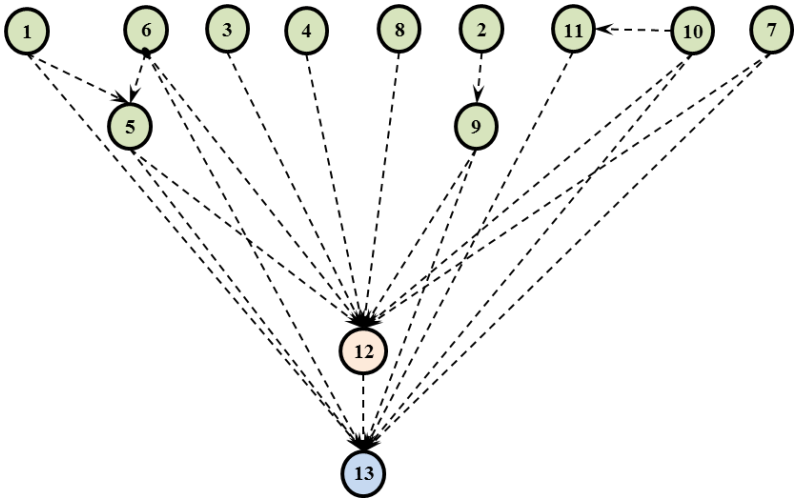

Figure 6.8: The expert elicited manual structure for the 13-variable LUCADA BN.

Following our elicitation of the manually built structure, we also collected pairwise relational information of different types from our expert panel in order to use for hybrid learning with CaMML. The notations for the types of structural pairwise priors are explained earlier in Section 6.3.2.2.3. As can be seen in Figure 6.9, the direct causal influences (A→B) of the pre-treatment variables on the treatment selection and treatment outcome variables are also prevalent in this pairwise-relations matrix. However, the flexibility of defining additional relation types as undirected relations (A−B) and temporal orders (A≺B) yields a slightly different view of the domain.

	1	2	3	4	5	6	7	8	9	10	11	12	13
1		≺	≺	−	−	≺	≺	≺	≺	≺	≺	≺	→
2			−	−								→	≺
3		−		−								→	≺
4												→	→
5				→								≺	→
6											→	≺	→
7												≺	≺
8									−				
9												→	→
10											→	→	→
11												→	→
12													→
13													

Figure 6.9: Expert elicited structural pairwise relations based on the selected 13 guideline variables. The variable codes are as given in Table 6.4. The notations can be read as: "A≺B": A happens before B; "A−B": A and B are related; and "A→B": A influences B.

The results of our experimental runs, which reflect the Bayesian scores and predictive performances achieved by different learning approaches, are as given in Table 6.5. If we initially focus on the average AUC and predictive accuracy values, we can see that there is not a significant difference between the performances of our baseline benchmarking algorithms and the BNs.

The two exceptions to this are 1) the decision tree learned by the C4.5 algorithm, which achieves a low AUC value relative to all other algorithms and 2) the manually constructed BN structure as given in Figure 6.8. The low performance of the manual DAG structure may be explained by the implicit dependencies within the data that the clinically elicited network is unable to capture.

	AUC	Accuracy %	Log Bayesian Score DAG_{final}
Logistic Regression	0.812 (±0.04)	77.00 (±0.61)	-
C4.5	0.767 (±0.04)	75.64 (±0.59)	-
Naïve Bayes	0.793 (±0.04)	75.04 (±0.67)	-
Tree augmented Naïve Bayes	0.810 (±0.05)	76.93 (±0.63)	-1,558,894
BN, IC	0.793 (±0.04)	75.04 (±0.74)	-1,763,061
BN, K2	0.809 (±0.04)	76.49 (±0.69)	-1,590,874
BN, Simulated Annealing	0.807 (±0.04)	76.50 (±0.71)	-1,567,118
BN, MCMC	0.807 (±0.05)	74.24 (±0.54)	-1,600,891
BN, CaMML - no priors	0.806 (±0.03)	73.10 (±0.73)	-1,586,574
BN, CaMML - temporal tiers	0.806 (±0.03)	74.31 (±0.72)	-1,570,878
BN, CaMML - structural priors	0.805 (±0.03)	74.27 (±0.64)	-1,581,243
BN, manually built structure	0.749 (±0.03)	68.30 (±0.62)	-2,093,036

Table 6.5: The predictive performance metrics and Bayesian Scores for the 10-fold stratified cross validation experiments with the corresponding algorithms. The AUC and Accuracy % columns represent the means and standard deviations of the cross validated results

From a causal perspective, it is remarkable that despite its strong independence assumptions, the Naïve Bayes (NB) classifier, "*long a favourite punching bag of new classification techniques*" [258], yields comparable results to the more sophisticated structure learning algorithms. While not suitable for use in causal inference scenarios,

Logistic Regression also performs on par with the BN structures learned on performing observational inference.

Focusing on the Bayesian scores achieved by different learning approaches, it is evident that the manually elicited structure attains the lowest score. Furthermore, the structure that obtains the highest Bayesian score among others is the one learned via the tree augmented Naïve Bayes (TAN) algorithm in Figure 6.10. This DAG, which does not have much resemblance to the expert-elicited structure in Figure 6.8, has the 1-year survival (13) node as its root node. Despite the fact that TAN is categorised under the structure learning algorithms in Section 6.3.2.2, as a slight relaxation of NB and, it is not intended for causal discovery. It is therefore quite surprising for this structure to attain the highest Bayesian score.

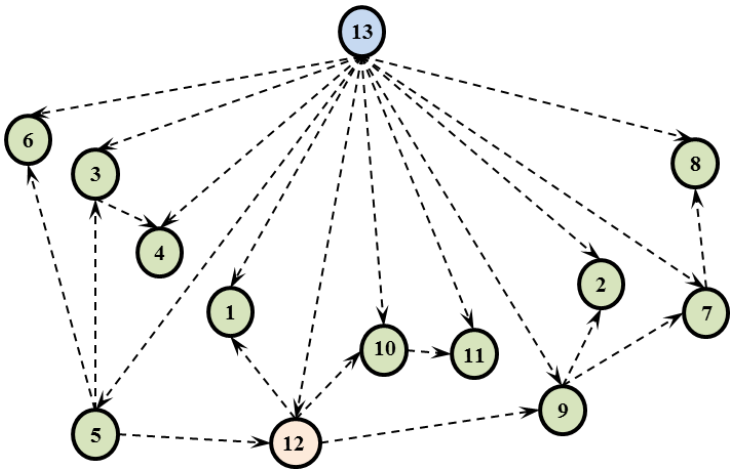

Figure 6.10: The DAG structure learned by the TAN algorithm.

Overall, all structures learned by score-based algorithms outperform the one which has been learned via conditional independencies in the dataset. It should, however, be kept in mind that in searching the domain of all probabilistic models, most score-based algorithms given in Table 6.5 aim to maximise the Bayesian score and consequentially it may not be surprising that the final Bayesian scores achieved by these algorithms are better than those of the constraint-based ones.

However, it is remarkable that despite operating on a different metric score, namely MML, the structures learned by CaMML achieve comparable Bayesian scores to the other score-based searching algorithms. Among the three CaMML experiments that use 1) no priors; 2) temporal tiers information with a confidence of 1.0 (as colour-coded in Table 6.4); and 3) structural pairwise relations (Figure 6.9) with confidences of 0.8, we can see that while the incorporation of expert knowledge into the learning process has little effect on the Bayesian score or the predictive performances attained, they help yield structures that look more similar to the expert elicited one given in Figure 6.8.

Figures 6.11 and 6.12 give the causal structured learned by CaMML with the added expert knowledge of structural pairwise relations and temporal tiers respectively. A comparison of the manually built structure (Figure 6.8) to these hybrid structures learned by CaMML reveals that the expert elicited DAG is less connected but has a higher max fan-in (especially for the treatment selection (12) and survival (13) variables) compared to the hybrid structures learned by CaMML.

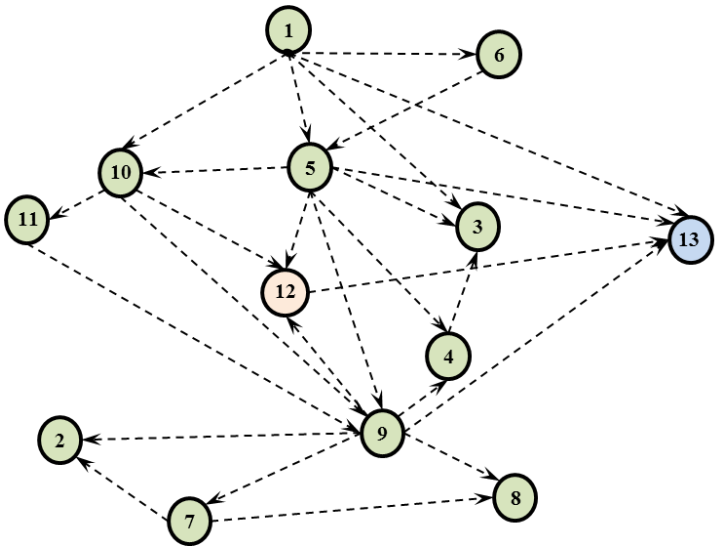

Figure 6.11: The hybrid structure learned by CaMML, specifying the structural priors as given in Figure 6.9 with a confidence level of 0.8.

As set out in the beginning of the chapter, our primary goal in running these experiments was to uncover the most feasible causal structure that would constitute our probabilistic

domain representation, analogous to the rule-based representation of the domain as given in Chapter 5.

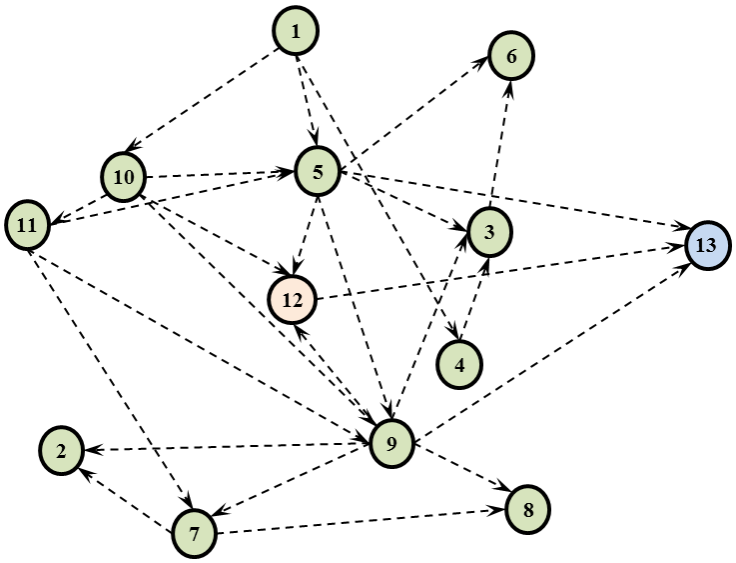

Figure 6.12: The hybrid structure learned by CaMML, specifying the temporal tiers as colour-coded in Table 6.4 with a confidence level of 1.0.

As a result, the structure we chose needed to be causally plausible, i.e. not openly violate the causal understanding of the domain as perceived by the clinicians. Therefore we opted to pick the causal structure learned based on the CaMML temporal tiers (Figure 6.12), which achieves the highest Bayesian Score among other structures that take into account expert knowledge and therefore successfully captures the primary causal patterns between the pre-treatment, treatment selection and treatment outcome variables.

6.5. Discussion

Our choice of BNs for providing causal inference was motivated by the clinicians' demand for a tool that can assist them in assessing the survival probability and how the selection of different treatment plans would affect it. Addressing this need requires a model that is capable of using interventional as well as observational inference for which BNs are well-suited.

In our experimental results, we attempted to establish two evaluative standards, namely the Bayesian scores calculated for the DAGs output by each algorithm and the corresponding

predictive performances on 1-year survival based on these DAGs. The results presented in Section 6.4 are unfortunately not sufficient to make a definitive statement on which structure learning methodology is the best since different methodologies prioritise different factors, e.g. various metrics, statistical dependencies, expert knowledge, for causal discovery. Nearly every structure learning publication in the literature makes some kind of empirical case in support of their proposed algorithm /methodology.

As discussed in the beginning of Section 6.1.2, the inability of the structure learning algorithms to yield a single DAG that faithfully represents the causal structure of a given domain is a well-known issue in causal discovery. As presented in detail in Chapters 8 and 9 of [115], "*many causal discovery algorithms evade, or presuppose some solution to the problem of identifying a correct variable order*". As the careful reader will note, there is actually a very substantial difference between the DAGs learned by CaMML and TAN, whereby the arc directions in the highest scoring TAN algorithm actually defy any temporal or causal pattern within the domain. On the other hand, we explained in Section 2.4.2.2 that manual construction of BNs is actually very common. However, our experimental results indicate that the manually constructed BN in our case does not represent the best model in terms of predictive performance or fit to data.

In practice, both automated and manual constructions of DAGs have limitations. An alternative to these, i.e. the hybrid causal learning, is an emerging field and shows promise in obtaining causal structures that yield high performance metrics while retaining the causal patterns set out by domain experts. Our empirical results reveal that the DAGs learned by the hybrid learning algorithm CaMML provides the highest Bayesian score while adhering with the causal constraints expressed by the domain experts.

To date, we have experimented just with the most prominent structure learning algorithms. Our initial results have given sufficient encouragement to conduct more extensive experiments in the future. In the next chapter, we will look into the causal intervention

results that convey information on how different treatment plans affect the probability of 1-year survival for a given patient.

Chapter 7 - Evaluation of LCA version 2

In Chapter 6, we selected the DAG, learned by the CaMML hybrid causal discovery software, to form the structure of the BN to integrate into Lung Cancer Assistant (LCA). During our evaluation of different DAG candidates, we described how well each algorithm performed in making personalised 1-year survival predictions. This, in effect, served as an evaluation of the system's performance in accurately answering the observational inference query of "$P(Survival = Alive \,|Evidence) = ?$", which we argued is vital in arriving at informed treatment selections decisions at the MDT meetings.

In this chapter, we continue with the empirical evaluation of our selected BN, this time assessing its performance in making plausible treatment recommendations based on the interventional query of "$P(Survival = Alive \,|Evidence, T) = ?$", introduced in Section 6.1.2. We run this experiment on the same subset of patients that we used for assessing guideline rule-based treatment recommendations given in Chapter 5. This facilitates a direct comparison between the treatment recommendations achieved by causal interventions and guideline rule-based decision support.

Following our discussion of results, we present the software architecture of the second version of LCA and provide details on the implementation of its design. Finally, we present our preliminary work on a technique that would allow automatic parsing of guideline rule knowledge in the SNOMED-CT LUCADA ontology into BN queries to provide quantitative degrees of support for the individual guideline rule-based arguments based on survival odd ratios.

7.1. Probabilistic Treatment Recommendation Results and Discussion

Before presenting the probabilistic treatment recommendation results, it is worthwhile to investigate: 1) the probabilities of treatment plans: $P(Treatment)$; and 2) the conditional

probabilities of 1-year survival when a specific treatment plan is given: $P(Surv = Alive\,|Treatment)$, as observed in LUCADA. Figure 7.1 shows these treatment frequencies and the 1-year-survival probabilities conditional on treatment plans as blue and green columns respectively for each treatment plan. The lack of correlation between the treatment frequencies and conditional survival probabilities in Figure 7.1 may reflect the fact that survival maximisation is not the only parameter affecting the eligibility of patients for a particular treatment plan. We elaborate on this further while discussing our results.

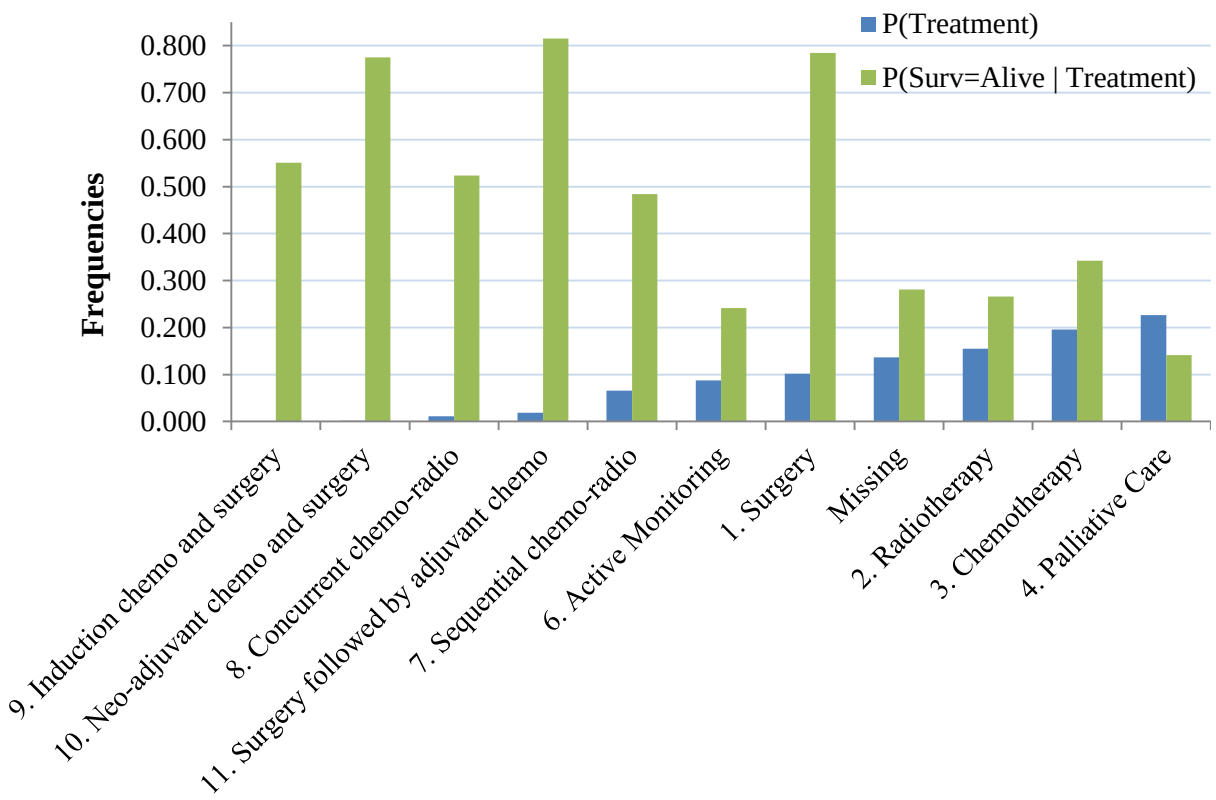

Figure 7.1: The treatment plan probabilities, $P(Treatment)$, represented in blue columns, and the conditional 1-year survival probabilities given specific treatment plans, $P(Surv = Alive\,|Treatment)$, represented in green columns, as calculated from LUCADA.

The discrepancy between $P(Treatment)$ and $P(Surv = Alive\,|Treatment)$ is more prominent in treatment plans that involve surgery (1, 9, 10, 11). For instance, if we focus on "Surgery followed by adjuvant chemotherapy", we may observe that $P(Surv = Alive\,|Treatment = 11) = 0.81$, while $P(Treatment = 11) = 0.02$. This means that despite the high chances of survival if given the treatment, the joint probability, $P(Surv = Alive, Treatment = 11)$, of observing a patient, who has been

given "Surgery followed by adjuvant chemotherapy" and survived at least one year, is at a meagre $0.016 = (0.81 \times 0.02)$ in the database.

7.1.1. Causal Intervention Concordance Results

We ran our experiments on the same patient subset described in Section 5.2.3.1. Since our motivation was to make personalised treatment recommendations on the basis of survival maximisation, we excluded the non-curative treatment plans: 'Active Monitoring' and 'Palliative Care' from our interventions. Before running the causal interventions, as recommended by Pearl [231], we modified the DAG structure given in Figure 6.12 by removing the edges directed at the intervened variable, "Suggested Cancer Treatment Plan". This modified DAG is shown in Figure 7.2.

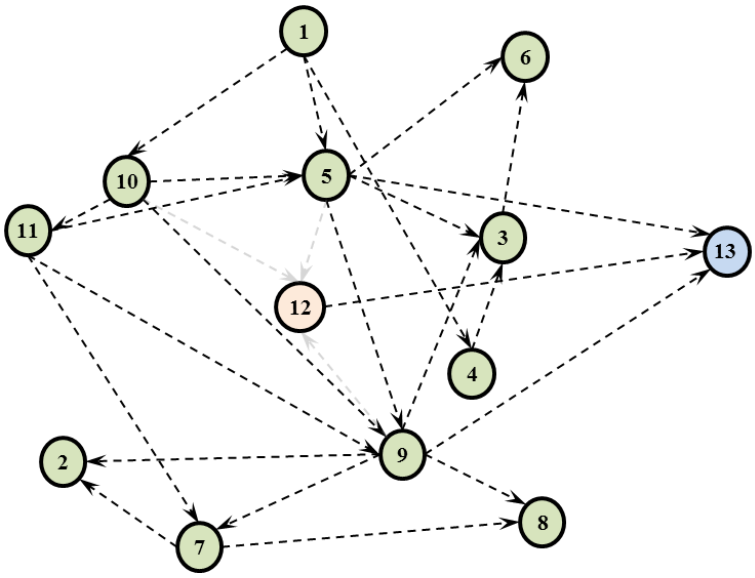

Figure 7.2: The BN Structure used for carrying out causal interventions on the "Suggested cancer treatment plan" variable.

While re-parameterising this new BN, we excluded the validation patient subset in order to minimise bias in our results. Similar to our guideline rule-based treatment recommendation experiments in Chapter 5, we evaluated the suitability of the BN recommendations with respect to both exact and partial matches between the top system recommendation that maximises survival and the recorded treatment in LUCADA. Overall, the percentage of patients for whom there was an exact concordance between the top LCA BN

recommendation and the recorded treatment was 29%. This percentage rose to 76% when we included partial matches between the two.

Concordances with respect to recorded cancer treatment plans

We first analysed the level of exact and partial concordances with respect to the recorded treatment plan types in the dataset. Figure 7.3 shows the confusion matrix summarising the aggregated discrepancies between the recorded treatment plans in the dataset and the top recommendations provided by the LCA BN. The cells, highlighted in orange, represent the most prevalent sources of discordance between the two.

		Top Recommended Plan							
		1.Surgery	2.Radiotherapy	3.Chemotherapy	7.Sequential chemo-radio	8.Concurrent chemo-radio	9.Induction chemo and surgery	10.Neo-adjuvant chemo and surgery	11.Surgery and adjuvant chemo
Recorded Plan	1.Surgery	681	0	0	5	0	63	148	1319
	2.Radiotherapy	129	0	0	9	0	47	96	292
	3.Chemotherapy	8	0	0	0	0	67	100	62
	7.Sequential chemo-radio	11	0	0	12	0	43	142	99
	8.Concurrent chemo-radio	4	0	0	1	6	6	42	41
	9.Induction chemo and surgery	0	0	0	0	0	2	7	7
	10.Neo-adjuvant chemo and surgery	0	0	0	0	0	2	21	24
	11.Surgery and adjuvant chemo	75	0	0	3	0	16	82	348

Figure 7.3: The confusion matrix that displays the recorded treatment plans versus the top probabilistic recommendations by LCA.

A clearly visible pattern in Figure 7.3 is that the top treatment recommendations by the LCA BN almost exclusively comprise surgery (labelled as 1, 9, 10, and 11). If we focus on the non-surgical treatment plan columns (labelled as 2, 3, 7, and 8) we see that the single modality plans: radiotherapy and chemotherapy are never recommended by the system,

and the multimodal chemo-radiotherapy plans are recommended very rarely. This is a major weakness of the current BN recommendations and will be discussed in detail in Section 7.1.2.

Focusing on the 'Surgery' row in Figure 7.3, we see that for the majority of the cases, the BN favours multimodality surgical treatment plans: 9, 10, and 11 over surgery alone. Analysing the characteristics of the 681 concordant cases, we found that these were all early stage patients, for whom surgery alone yielded comparable or marginally better survival expectancies compared to the multimodal surgical plans. Another interesting observation here is that the treatment plans 9 and 10 are on-going clinical trials and are only given to a limited number of patients for the time being. As can be seen in the confusion matrix, based on maximising the probability of 1-year survival, the BN recommends these plans for a comparatively significant number of patients.

In addition to the confusion matrix in Figure 7.3, we also provide a stacked column graph that summarises the exact and partial concordances with respect to different treatment plan types. Concentrating on the non-surgical treatment columns in Figure 7.4, we can observe that they are mostly comprised of discordant cases.

It is clear both from Figures 7.3 and 7.4 that the maximum a posteriori (MAP) estimations of $argmax_T[P(Survival = Alive\,|Evidence, T)]$ produce recommendations that are heavily biased towards surgical treatment plans. For this reason, we ran a second set of experiments in which we only included those patients from the selected subset for whom the recorded treatment plan was non-surgical. Furthermore, we excluded surgical treatment plan types 1, 9, 10 and 11 from our interventions, to assess whether the concordance levels improved when we manually eliminated surgical treatment plans as viable options.

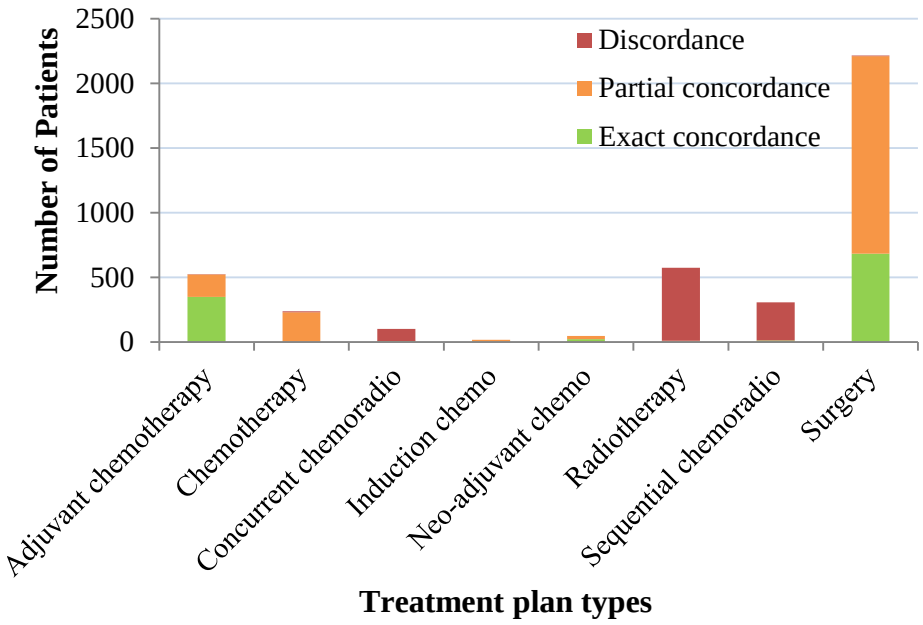

Figure 7.4: The exact and partial concordances between the system recommendations and the recorded treatment plans stratified with respect to different treatment plan types.

Presented in Figure 7.5, these results reveal that when surgical treatment plans are discarded, the concordance levels between the system recommendations and recorded treatments increase substantially.

		Recommended Plan			
		2.Radiotherapy	3.Chemotherapy	7.Sequential chemo-radio	8.Concurrent chemo-radio
Recorded Plan	2.Radiotherapy	161	12	172	228
	3.Chemotherapy	10	2	113	112
	7.Sequential chemo-radio	13	1	128	165
	8.Concurrent chemo-radio	4	0	41	55

Figure 7.5: The non-surgical confusion matrix for patients who have been treated with no-surgical treatment plans. The columns represent the non-surgical recommendations.

When we investigated the characteristics of the concordant and discordant cases on the 'Radiotherapy' row, we saw that the 161 concordant cases are all early-stage (IA and IB) cancer patients, for whom the 1-year survival probabilities achieved by 'Radiotherapy'

alone are comparable or greater than the multi-modal treatment plans 7 and 8. However, from stage IIA and upwards, the BN recommendations heavily favour multimodal chemo-radiotherapy treatment plans 7 and 8 over radiotherapy alone.

Upon further analysis of the posterior survival distributions for the patients, who were recommended either sequential or concurrent chemo-radiotherapy plans (7 and 8) by the system, we saw that in most cases the 1-year survival probabilities with either treatment plan were very similar, slightly varying in favour of 7 or 8 depending on patient characteristics. The system's indecisiveness in distinguishing between these two plans may be indicative of additional criteria, other than maximising survival, that affect this decision in real life. Finally, a striking observation in Figure 7.5 is that, apart from 2 patients (who were both stage IIA), the causal non-surgical treatment plan interventions on the BN never resulted in 'Chemotherapy' being the treatment that maximises survival. As is visible, in the majority of such cases, the system consistently favoured the multi-modal chemo-radiotherapy plans over chemotherapy alone.

Concordances with respect to TNM Stages

We also investigated the levels of exact and partial concordances with respect to the TNM stages of the test patients, as given in Figure 7.6.

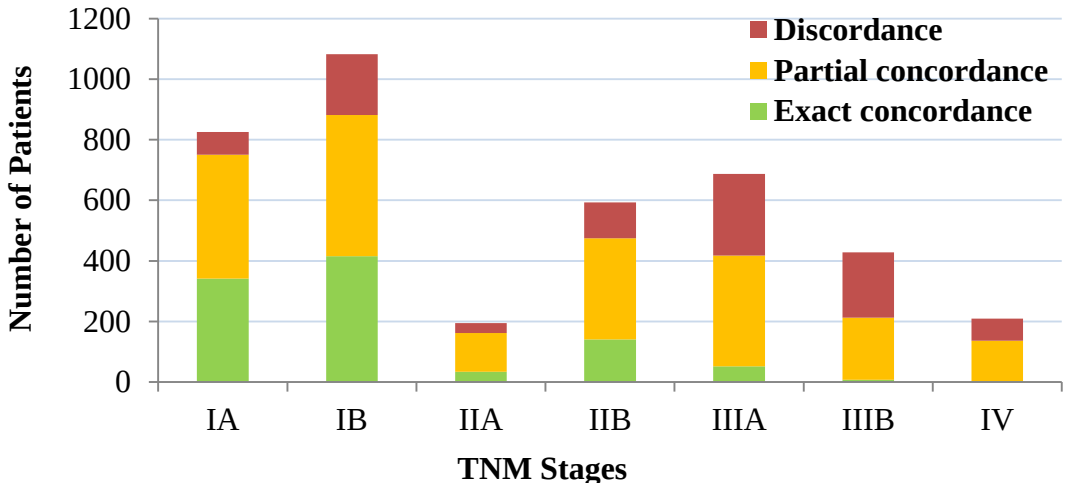

Figure 7.6: The exact and partial concordances between the probabilistic recommendations and the recorded treatment plans stratified with respect to the TNM stages.

As can be seen, the exact concordance levels plummet for locally advanced (IIIA and IIB) and advanced (IV) stage patients. This is because the system consistently favours surgical treatment plans, whereas in clinical practice the proportion of patients who are suitable for surgery decreases as the TNM staging increases. This inverse correlation between the concordances and TNM staging can also be explained by the fact that the BN recommendations increasingly favour multimodal treatment plans over single modality ones as the severity of the disease increases.

7.1.2. Discussion on Causal Intervention Results

Our experimental results reveal that decision making based on maximising 1-year survival are highly biased and as such not very reliable in predicting recorded treatments. On the other hand, the posterior distributions represent a more complete picture, revealing not only what the treatment plan that maximises survival is; but also to what extent it improves 1-year survival expectancies relative to the alternative options. As an example, for locally advanced stage patients, a detailed investigation of the posterior distributions reveal that the added survival benefit of the surgical plans are not as significant compared to the multi-modal chemo-radiotherapy plans.

The main source of disagreement between the top system recommendations and recorded treatment plans stems from the discrepancies between the conditional and joint probabilities of 1-year survival as discussed in the beginning of Section 7.1. Due to the nature of our causal interventions, the system recommendations are based on the conditional survival probabilities, while the frequencies in the database reflect the joint probabilities $P(Surv = Alive, Treatment)$.

The lack of correlation between the two indicates that survival maximisation is clearly not the only parameter affecting the clinicians' choice of a treatment plan. In fact, there are various other factors that govern treatment selection decisions. The most prominent

examples to these are: 1) the suitability of the patient for surgery or other treatment modalities; 2) the quality of life evaluation during and after treatment; and 3) an economic analysis on the cost efficiency of the treatment plans. These all directly contribute to the treatment decisions that comprise the treatment frequencies in LUCADA.

On the other hand, the causal intervention queries on the LUCADA BN do not take these additional factors into account. This is primarily due to lack of available data on these factors in LUCADA, which disallow us from carrying out multi-criteria decisions analyses, an example of which could have been causally intervening on additional variables such as 'suitability for resection (R)' and 'cost effectiveness (C)', alongside 1-year survival (S) and return posterior distributions reflecting: $P(S = Alive,\ R = Yes, C = High\ |Evidence, T)$. As we will discuss further in the final chapter, such utility-based analyses can be carried out with decision networks, which are generalisations of BNs with added functionality that allow multi-criteria decision analysis.

7.1.2.1. Suitability for Resection

As the definitive treatment for lung cancer, it is well-established that surgical resection provides the highest probability of survival [281]–[283]. A recent study by Riaz et al. concludes that the low survival rates in the UK can partly be attributed to low and variable resection rates across primary care trusts [284]. Evidence from NLCA and the Society of Cardiothoracic Surgeons show that although the curative resection rates in the UK are rising, they are still not at the NLCA's recommended rate of 52% [285], [286].

These highlight that the surgery decision is extremely vital in lung cancer care. As a result, the inability to evaluate "suitability for surgery" seems to be the major weakness of the LUCADA BN recommendations. As covered in Chapter 5, according to the BTS and NICE guidelines, suitability for surgery should be determined by factors such as: risk of peri/post-operative mortality, cardiac functional capacity, lung function, and post-operative

quality of life [3], [7]. Unfortunately, we cannot incorporate these into our probabilistic queries since relevant information is not stored in LUCADA.

However, this limitation can be practically addressed by displaying the posterior survival probabilities for surgical and non-surgical treatment plan recommendations separately. This would make it clear to the clinicians that LCA is unable to aid in evaluating suitability for surgery, while explicitly presenting the benefits of surgical treatment plans on survival expectancy. This may, in effect, highlight the significant benefits of surgical treatment plans at the MDT meetings. Resection rates are reportedly higher when the patient is discussed in an MDT meeting preoperatively [287], [288]. In this respect, adoption of LCA, which displays the survival expectancies with and without surgical interventions, can help increase the number of resection decisions further at the MDT meetings.

7.1.2.2. Limitations

It should be kept in mind that due to time and data availability constraints, our experiments have some limitations. First, due to the lack of information on 5-year survival rates, in our experiments we adopt a surrogate outcome measure, namely 1-year survival. Despite the fact that our choice of the 1-year survival surrogate outcome measure is not arbitrary [163], [164], it is possible that the probabilistic treatment recommendations may change if the gold standard 5-year survival rates are used.

Second, LUCADA, upon which we trained our BN and ran retrospective experiments, contains treatment decisions which were supposedly made by following clinical guideline rules. As such, it reflects biased treatment patterns. Unfortunately, the only systematic way of circumventing this would be by using data collected during prospective pilot studies, which span over a minimum of 5-years and ideally involve randomised control groups.

Finally, it can be argued that the 4020-strong ‘validation subset’ should have been excluded from our structure learning experiments from the beginning. While we are aware

of the potential biases this may introduce in our results, our choice to not exclude these patients from structure learning was motivated by the fact that these constituted the most complete patient subset in LUCADA. In addition, since our causal discovery was not solely based on data, the potential biases introduced may be further reduced.

7.2. Lung Cancer Assistant version 2

We integrated the probabilistic inference functionality into LCA in order to provide quantified and patient-specific answers to the elementary questions on the probability of survival and selection of the treatment plan that would maximise the chances of survival. These necessitated making use of the BN structures given in Figure 6.12 and Figure 7.2 for running observational and interventional queries respectively.

7.2.1. System architecture

The major difficulty in realising the design was programming additional classes within the LCA GWT project that would allow: 1) Creating BNs based on the DAG structures given in Figures 6.12 and 7.2; 2) Learning the parameters of the BN based on LUCADA, which is stored within a PostgreSQL database; 3) Observe evidence based on patient records in the database and perform probabilistic inference on the BNs in order to return posterior marginal distributions of the variable(s) of interest.

For creating generic representations of the BN structures, we made use of the online JavaBayes applet [289], developed by Fabio Cozman. JavaBayes allows building and saving BNs in the standardised Bayesian Interchange Format (BIF) [290] that is compatible with the majority of commercial and educational BN software tools. BIF provides a well-structured and standardised XML representation of the DAG and parameters of a BN. For parameterising the BNs stored in the BIF format, we wrote a program that loops through all variables in the BN and automatically generates SQL queries from templates to acquire the corresponding CPTs, as calculated by equation 6.5 in

Section 6.3.2.1. Similar to our MatLab and WEKA experiments, we used uniform Dirichlet priors with weights of '1' in populating the CPTs for all variables. Finally, we made use of the bucket tree algorithm, described in [291] and implemented by Cozman in EBayes [292], as our BN inference engine. EBayes is a freely available and embedded library of JavaBayes that allows belief updating on discrete BNs stored in BIF and XMLBIF formats.

We packaged all BN functionalities explained above within a "Bayesian Worker" Java class on the server side. Figure 7.7 shows the resulting system architecture of the second version of Lung Cancer Assistant.

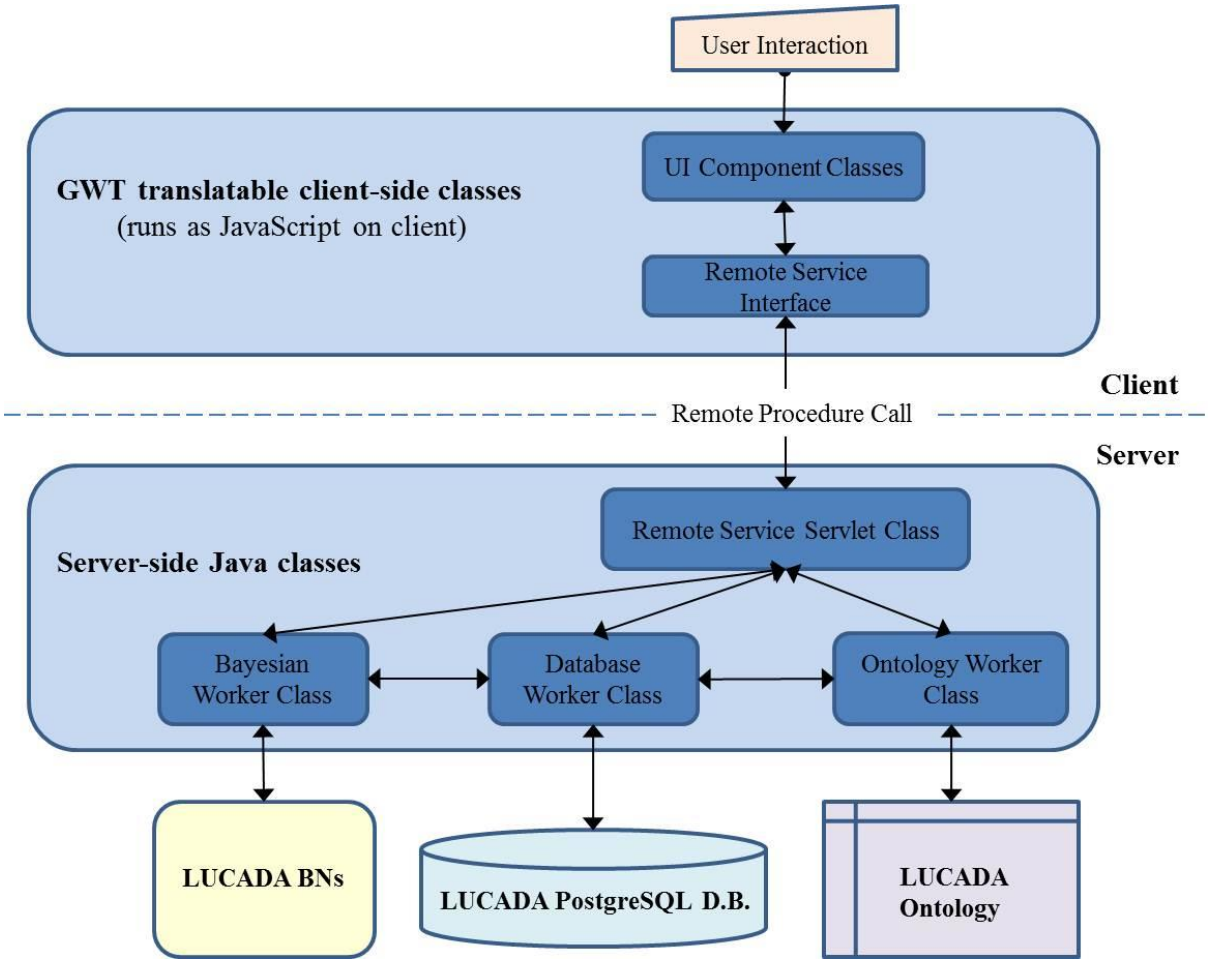

Figure 7.7: The software architecture of the GWT-based Lung Cancer Assistant CDS prototype.

As the reader may observe, the architecture given in Figure 7.7 is very similar to the architecture of the first version of LCA, given in Figure 5.1. The major difference is the addition of the "Bayesian Worker" Java class which integrates all aforementioned

probabilistic inference related operations into the system architecture. As can be seen, the "Database Worker" Java class takes on a central role, enabling the communication of electronic patient information with both "Ontology Worker" and "Bayesian Worker" classes.

7.2.2. Realisation of Design

We incorporated the probabilistic decision support functionality into the 'Decision Support' tab of the user interface, which initially only contained rule-based recommendations. According to this, the right half of the screen is allocated to displaying the interactive graphs that inform on the personalised probability estimations provided by the observational and interventional queries on the LUCADA BN. The cumulative inference times taken by the system to run the observational and interventional queries is measured in milliseconds, which complies with our objective of providing instantaneous feedback to the clinicians upon patient data entry.

As can be seen in Figure 7.8, the 1-year survival probabilities conditional on different curative treatment plans are displayed within a bar chart at the top right corner. For reasons explained in Section 7.1.2, the probability bars that correspond to surgical and non-surgical treatment plans are colour-coded differently. In addition, the personalised 1-year survival probability estimation is displayed within a pie chart at the bottom right corner. Combined with the guideline rule-based recommendations displayed on the left, the additional information provided by these graphs convey a more complete picture of the patient being discussed.

Focusing on the example case displayed in Figure 7.8, we see that the personalised guideline rule-based and probabilistic recommendations both indicate "Surgery followed by adjuvant chemotherapy" as the most favourable option for the 55-year old patient.

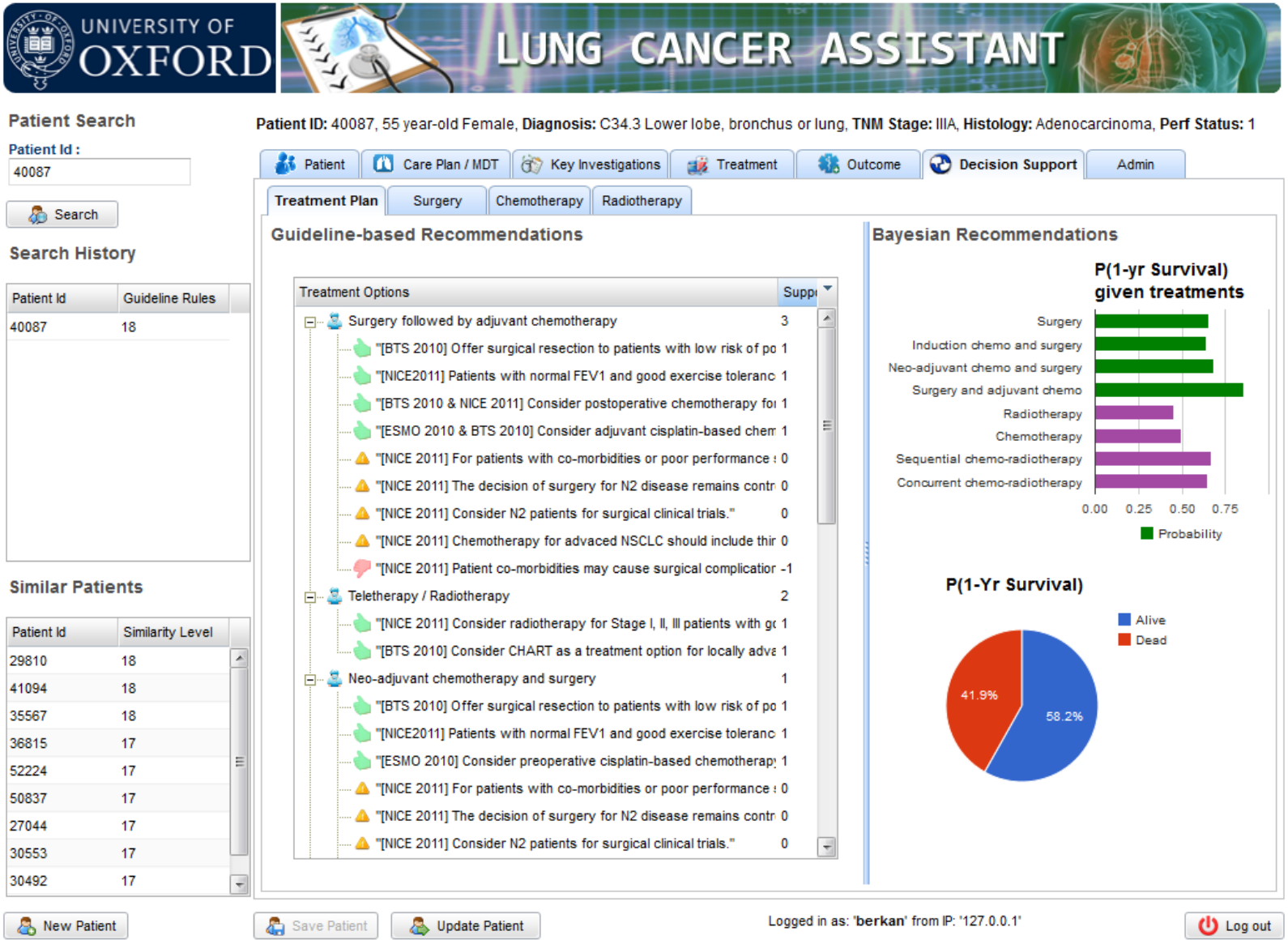

Figure 7.8: Screenshot of the LCA 'Decision Support' tab.

From the pie chart, we can see that the probability of this patient surviving a year is approximately 58%. It is worthwhile mentioning that this probability is irrespective of treatment selection and reflects the survival expectancy conditional on the rest of the variables in the BN. Meanwhile, the causal intervention results reveal that, given "Surgery followed by adjuvant chemotherapy", the 1-year survival probability of the patient would significantly increase to 86%.

A complementary function of the probabilistic recommendations here is that the two graphs displayed on the right present results for all patients, irrespective of the treatment intentions being curative or palliative. Given all relevant sources of information, a clinician can make use of the explicitly displayed posterior distributions in many ways. For instance, though of course this example raises uncomfortable ethical questions, a low 1-year survival probability within the pie chart may be a counter-indicator to prescribing expensive treatment plans that have poor expected outcomes.

7.3. Comparison of Rule-based and Probabilistic Recommendations

In this section, we compare the most notable characteristics of the guideline rule-based and probabilistic treatment recommendations presented to this point in the thesis. A direct comparison of the empirical concordance results between the two reveals that rule-based decision support, which achieves exact and partial concordance percentages of 57% and 79% respectively, is better in simulating the recorded treatments in the database. As underlined in Section 5.3, while a high concordance rate with the recorded treatments does not necessarily imply better decision support, the rule-based CDS results at least provide sufficient evidence that the system is capable of making sensible personalised recommendations.

On the other hand, the relatively poorer concordance results of the probabilistic recommendations can be explained by the inability to incorporate additional factors, other than maximising survival expectancy, into our causal interventions. While, it is very important to emphasise that this shortcoming is not methodological but is merely due to lack of data on such factors in LUCADA, it is obvious that the usefulness of the MAP estimates on treatment recommendations provided on the basis of survival maximisation is very limited. However, the posterior distributions are very informative in allowing the clinicians to compare the direct impacts of different treatment plans on survival expectancies. Of particular importance is the ability they provide to highlight the significant benefits of surgical treatment plans on survival, which may assist in increasing the rates of surgical care plan decisions.

The relatively better concordance results achieved by the guideline-rule based CDS is due to their qualitative nature, which allows them to implicitly accommodate factors other than survival maximisation in their recommendations. Therefore, in the absence of comprehensive electronic patient data, guideline rule-based CDS clearly serves the important purpose of laying out a more complete picture of the factors that govern treatment selection decisions. However, on the flipside, more often than not the qualitative nature of the guideline rules manifests itself in the form of vagueness and uncertainty in rule eligibility criteria and can therefore be highly detrimental.

Comparisons of Empirical Results based on Treatment Plan

A treatment selection pattern observed in both guideline rule-based and probabilistic recommendations is the relative over-prescription of surgical treatment plans compared to recorded clinical practice. Unless there are strong contraindications, such as metastatic disease, poor performance status or low lung capacity, the rule-base of LCA prioritises

single or multiple modality surgery plans for early and locally advanced cancer patients. We would argue that the comparative over-prescription of surgery by the guideline rule-based recommendations is justifiable at large. The 'over-prescription' of surgery by the probabilistic decision support, on the other hand, is caused by the limitations of the one-dimensional analysis based on survival maximisation. However in either case, the fact is that resection rates in the UK are lower than desired levels and hence there should be no obvious harm in laying out objective qualitative and quantitative indicators in support of surgical treatment plans at the MDT meetings.

Another joint pattern that we observed in both rule-based and probabilistic recommendations was that compared with recorded practice, they both favour multi-modality treatment plans over single modality ones. The most significant example in this respect is the single modality chemotherapy treatment plan, which is very rarely recommended by guideline-rule based CDS and almost never (apart from two patients) recommended on its own by the probabilistic CDS.

In addition, we also observed that guideline rule-based CDS never recommends the clinical trial treatment plans (9 and 10) due to lack of evidence in the literature. In contrast, the probabilistic recommendations tend to favour these treatment plans based on survival maximisation. It should, however, be noted that the patient subset, who have been treated with treatment plans 9 or 10, is relatively small.

Another point to note is that neither the guideline rule-based nor the BN recommendations can clearly distinguish between the sequential and concurrent chemo-radiotherapy cases. In Chapter 5, we reported that for guideline rule-based recommendations, this may stem from the lack of clear guideline definitions of "suitable for sequential but not suitable for concurrent chemo-radiotherapy". In the causal intervention results presented in this

chapter, the inability to distinguish the two most likely arises from the fact that, just like with surgery decisions, the suitability for concurrent chemo-radiotherapy concerns other criteria than survival maximisation, which are not included in the BN.

Apart from treatment recommendations, an added benefit of incorporating probabilistic recommendations into LCA was the ability to view personalised 1-year survival expectancies, which may significantly affect the treatment decisions of the MDT. Also, in contrast with guideline rule-based CDS, the probabilistic inference is not limited to patients suitable for curative treatments but is available for all patient types.

Finally, another advantage of the probabilistic inference provided by the BNs is that the probability distributions underlying the network can be automatically updated to incorporate newly added patient information. In Chapter 5, we had reported that the maintenance of rule-bases by clinicians and informaticians is a major drawback in the wide adoption of these systems. In contrast to this dependency, the adaptive nature of the BNs provides a more autonomous system that can evolve as more data is added. Recently, this has led to the adoption of BNs in building adaptive management models especially in ecological and environmental domains [293] .

Chapter 8 - Conclusions and Future Directions

In this thesis, we have presented the design and implementation stages of a novel CDS application, Lung Cancer Assistant (LCA), which integrates rule-based and probabilistic inference in order to aid clinicians to arrive at more informed treatment selection decisions in the lung cancer MDT meetings.

Our research was motivated by the increasing clinical need for CDS in MDT meetings and the limitations of the conventional rule-based CDS applications. Through our attendance at the lung cancer MDT meetings at the Churchill Hospital and frequent meetings with our clinical collaborators over the course of the project, we have formed a good understanding of the major requirements of a CDS prototype that would facilitate the decision making of the clinicians at the MDT meetings. LCA is the outcome of our efforts to satisfy these requirements within a single software platform.

8.1. Discussion of Findings

Due to the lack of a suitable publicly available computer interpretable guideline formalism that we can build upon, we implemented an ontological guideline rule inference framework that operates on an automatically extracted SNOMED-CT module integrated with our manually designed LUCADA ontology. We extended this ontology to incorporate guideline rule knowledge and utilised a semantic reasoner to infer guideline rule eligibility of patient records. The strength of this approach stems mainly from making use of a highly expressive semantic language (OWL-2) to model medical knowledge based on the standardised SNOMED-CT ontology.

In order to establish the connection between the patient records in the database and our domain ontology, we wrote a program that automatically translated a patient record into an OWL-2 T-box query. With the help of a semantic reasoner, we made use of these queries

to determine the guideline rules that apply to a patient record. As proven empirically, this approach enables a fast and scalable solution for ontological inference of guideline rule eligibility for patient records.

Furthermore, as a convenient feature of this framework, during ontological reasoning, patient records that satisfy semantically identical criteria are automatically grouped under the same guideline rule class definitions. This allows an intuitive notion of semantic similarity between patient records. In this manner, when a new patient is entered, the ontological inference makes it possible to display similar patients, which may inform treatment decisions.

After establishing the ontological guideline rule inference framework, we reviewed the major clinical guideline documents in lung cancer care in order to form the rule base for Lung Cancer Assistant. We worked closely with our clinical collaborators in interpreting and computerising these rules with respect to the variables available in the LUCADA data model. During our knowledge elicitation efforts; we found that representing the vague phrases in the clinical guideline rules constituted a major bottleneck. This slow and arduous process may indeed be the main obstacle in the wide-spread use of such rule-based expert systems.

After populating our rule base, we evaluated the plausibility of the rule-based system recommendations provided by LCA. For this reason, we measured the levels of exact and partial concordances between the system recommendations and the recorded treatment plans for a carefully selected patient subset in LUCADA. Our empirical results revealed that the rule-based inference was capable of making sensible recommendations, despite failing to closely simulate the treatment decisions for locally advanced and advanced stage

lung cancer patients, for which the level of uncertainty in terms of best practice is still very high.

However, the rule-based decision support functionality of LCA was incapable of providing answers to the clinically vital questions of the marginal survival probability of a given patient, and of how the selection of different treatment plans affects it. Motivated by the clinical demand, we extended LCA to incorporate probabilistic inference that could answer these two questions. For this purpose, we set out to design a Bayesian network (BN) whose DAG captured the causal structure between the domain variables, as elicited from the clinicians.

We based the selection of the BN structure to incorporate into LCA on these two factors: 1) causal plausibility; and 2) a high Bayesian score that represents the joint probability for a proposed DAG given data. We tested different BN learning algorithms on the LUCADA data and finally opted to select the DAG learned by the hybrid causal discovery algorithm CaMML. The BN learned from this structure had a relatively high performance in predicting the 1-year survival with an accuracy of 74.3% and an AUC of 0.806.

Following this, we utilised our BN to predict the personalised 1-year survival probabilities for patients given different treatment plans. In contrast with the observational survival prediction queries, returning the hypothetical posterior distributions of survival given different treatment plans required causal interventions on the treatment plan variable in our BN. After making the necessary modifications to our selected BN, we evaluated how closely the top recommendations of the BN followed recorded clinical practice, based on maximising 1-year survival. For this reason, we measured the levels of exact and partial concordances between the top system recommendations and the recorded treatment plans for the same patient subset used for evaluating rule-based recommendations.

We observed that compared to rule-based decision support, the BN recommendations, which were aimed at maximising 1-year survival, performed worse in simulating the recorded treatment decisions in the database. The recommendations produced in this way almost exclusively favoured surgical treatment plans over non-surgical ones. As we have extensively discussed, this was due to the additional factors -other than prolonging life expectancy- that govern treatment decisions, which were not included in the BN due to lack of data availability. This revealed that the maximum a posteriori (MAP) estimations achieved in this way were not very informative in predicting the recorded treatments.

However, this does not detract from the fact that an analysis of the posterior survival distributions conditional on treatment selections is highly informative in explicitly laying out not only what treatment plan maximises survival; but also to which extent it improves 1-year survival expectancies relative to the alternative options. This usage is also more compatible with our design goals since, as a CDS application, the primary function of LCA is to aid the clinicians in arriving at informed treatment decisions rather than to perform classification or MAP estimations and make that decision on their behalf.

Benchmarking the empirical results achieved by rule-based and probabilistic system recommendations, we found that both approaches tended to over-prescribe surgical treatment plans compared to clinical practice. This is in line with the efforts of the NLCA to increase resection rates across the UK. We observed that the relatively better concordance performance achieved by the guideline-rule based CDS is due to the qualitative nature of the guideline rules that allows them to implicitly accommodate factors other than survival maximisation in their recommendations. As a result, in the absence of comprehensive electronic patient data, guideline rule-based CDS clearly serves the important purpose of laying out a more complete picture of the factors that govern

treatment selection decisions. On the other hand, probabilistic inference provides a more precise means of decisions support that is also easier to build and maintain.

Overall, we believe the empirical results presented in the thesis are valuable not only in guiding the direction of similar future research in designing hybrid CDS systems that integrate probabilistic and rule-based CDS but also in providing a comparison of the strengths and weaknesses of probabilistic and argumentation-based methodologies in clinical decision support.

Satisfying practical concerns, such as fast inference speed and scalability, LCA is more than a proof of concept application. The potential of this tool has been recognised in the 11th annual British Thoracic Oncology Group conference in Dublin, where it won the best research prize among 216 abstract entries from all areas of thoracic oncology.

There are currently 21 clinicians from across the UK and Ireland who have requested and been given access to the online prototype available on http://lca.eng.ox.ac.uk. However, we are aware that a clinical adoption of the system requires extensive pilot studies, which can be carried out in the future, subject to the availability of research funding and continuation of clinical interest.

8.2. Future Directions

Despite its merits, as a research prototype LCA inevitably suffers from some limitations. Acknowledging these, in this section we propose ways to address some of these limitations, which constitute obvious avenues for future research.

8.2.1. Using the BN to quantify degrees of support for arguments

The integration of probabilistic and rule-based inference within LCA is practical rather than fundamental. One of the major limitations of the argumentation-based decision model

we adopted in LCA is the lack of meaningful and quantified indicators of support for the patient specific arguments displayed within the "Decision Support" tab, where we aggregate the support for an individual treatment plan by simply adding up the number of supporting arguments and subtracting the number of opposing ones.

Ideally, a quantitative and more precise measurement of the degree of support for an argument may be more informative and help to decide whether a hypothesis proposed by the arguments can be believed or not. In 2006, Williams and Williamson [112] published a proof-of-concept breast cancer prognosis application which presented a framework, combining logic with probability. While keeping the argumentation framework and the probabilistic components separate, they have demonstrated the added value of using posterior probabilities obtained by Bayesian inference to weigh up competing arguments in an argumentation framework. Although their work used rather simplified BN techniques, it was important in their conclusion that "*the number of arguments for and against a conclusion cannot be used in isolation to predict the likelihood of the outcome – some arguments count more than others and the probabilistic component provides a way to measure this.*"

For this purpose, we carried out an initial exploration on a methodology that would allow the automatic creation of BN queries based on the ontological guideline rule knowledge in order to appoint quantified support to rule-based arguments. We have developed an 'OWL-to-BN' Java class that can translate all defined 'Patient Scenario' equivalent class expressions into BN evidences that can be input to the 'Bayesian Worker' Java class described in Section 7.2.1. This functionality can be demonstrated on the OWL Class expression that represents (Rule 1) from Chapter 4.

"Patient and (hasPerformanceStatus some (WHOPerformanceStatusGrade0 or WHOPerformanceStatusGrade1)) and (hasClinicalFinding some (NeoplasticDisease and ((hasPreTNMStaging value "IIIA") or (hasPreTNMStaging value "IIIB") or (hasPreTNMStaging value "IV")) and (hasPreHistology some NonsmallCellCarcinoma)))" **(Axiom 1)**

The 'OWL-to-BN' Java class browses through all constituent axioms in a class expression and return all relevant object and data properties along with their values. For Axiom 1, it produces the output given in Figure 8.1.

```
---------- Rule 1 ----------
Eligibility:
hasTNMStaging equal [IV, IIIA, IIIB]
hasPerformanceStatus equal (hasRange) [WHOPerformanceStatusGrade1,
WHOPerformanceStatusGrade0]
hasHistology equal (hasRange) [NonsmallCellCarcinoma]
```

Figure 8.1: The output by the OWL-to-BN Java class

In addition to the eligibility criteria, for each guideline rule, the 'OWL-to-BN' class also returns all treatment plans recommended by the corresponding rule. We then parse the output of the 'OWL-to-BN' class into a BN evidence array that is compatible with the 'Bayesian Worker' class. Since the variables used in the guideline rule definitions and the BN are the same, mappings between the two are possible and we can return the probabilities of survival with the recommended treatment plan for the patient group described by the rule eligibility criteria.

An important thing to note here is that, due to the nature of clinical guideline rules, many variables will have to take on multiple states, which can be managed by re-parameterising the BN by dichotomising the corresponding variables to consist of two categories: one that satisfies the guideline rule criteria and one that represents its complement. For instance, in the example given here, the "TNM Category" variable would consist of: {TNMRule1, ¬TNMRule1} states, where TNMRule1= {IIIA, IIIB, IV}. Similarly the "Suggested Cancer Treatment Plan" variable would be dichotomised into: {Chemotherapy, ¬Chemotherapy}

states. After locally re-parameterising the relevant variable CPTs with respect to the guideline rule knowledge, causal intervention queries can be run to separately calculate the survival expectancies given the dichotomised states of the "Suggested Cancer Treatment Plan" variable. This can then be converted into a survival odd ratio for the given rule as shown in (8.1).

$$\frac{P(Surv = Alive|Evidence,\ \ Treatment = Chemotherapy)}{P(Surv = Alive|Evidence,\ \ Treatment = \neg Chemotherapy)} \quad \textbf{(8.1)}$$

The survival odd ratio can be interpreted as a measure of how significant of an impact adherence to the corresponding rule has on survival expectancy. This may, in turn, be used to quantify the level of support for the corresponding rule in LCA. The greater the survival odd ratio for a given rule, the more emphasis it should have compared to other arguments.

While the re-parameterisation of the BN for each rule can be performed by the 'Bayesian Worker' class, the mappings between the OWL class expressions and the BN variables remain problematic for discretised variables such as 'Age' and 'FEV1' measures. In addition, the conversion between the co-morbidity relations in the OWL class expressions and the "Number of comorbidities" variable would need to be manually established.

As a result, while the methodology described above would work seamlessly for half the guideline rules in our rule-base, for the other half manual mappings would have to be coded, hindering the generality of the solution. Furthermore, for reasons extensively discussed in Section 7.1, we are aware that evaluating the strength of the arguments solely based on survival maximisation would most likely result in incomplete and biased scores to be appointed to the arguments. However, in the presence of data on other factors that affect treatment decisions, the methodology described here can prove to be powerful in

assessing the validity of the guideline rule recommendations based on data and appointing consistent quantified support to the arguments.

8.2.2. Building a Utility-Based Model Using a Decision Network

As discussed in Chapter 7, due to absence of data on additional criteria, in its current version LCA bases its probabilistic recommendations on the posterior distributions of 1-year survival probabilities. In the presence of additional criteria that affect the ultimate treatment decision, a formalism that supports a multi-criteria utility model would be more suitable for representing the domain and the decision making process of the clinicians.

Bayesian networks are useful for large probabilistic inference problems such as those in medical diagnosis, survival prediction or causal interventions [294]. However, they do not allow for the evaluation of decisions or for representation of utility models. Decision networks (DNs), also known as influence diagrams, can be utilised to calculate the expected utilities of decision alternatives with respect to different utility indicators [295].

Within the context of the lung cancer treatment selection problem, the "Suggested Cancer Treatment Plan" variable can be represented as a decision node and the additional decision criteria such as 'health economic evaluation' or 'quality adjusted life expectancy' can be represented as value nodes that inform treatment selection in a DN.

This would of course be subject to the availability of data on such factors. As the granularity and coverage of information is improved within LUCADA, the scope of treatment recommendations can also be extended to include specific chemotherapy regimens or radiotherapy dosages.

8.2.3. Clinical Adoption and Pilot Studies

Despite the clinical interest in LCA, the results reported in this thesis are insufficient on their own to make a definitive statement on the positive impact of CDS provided by LCA in the MDT meetings. In order to provide a systematic and unbiased assessment on the impact of LCA in the clinicians' decision making, the system needs to be deployed in a prospective pilot study within a real clinical setting. This is a large commitment that would ideally span over a period of minimum 5-years with a randomised control arm built into the study design.

This would require further funding and a seamless integration of the software into the clinical workflow of the clinicians. In order to promote the use of LCA and facilitate its acceptance in the MDT meetings, additional functionalities that would help in the administrative management of the meetings can be added into the system. As discussed at the very beginning of the thesis, the main form of computer support in the MDT meetings is medical image analysis. Therefore it seems vital for a pilot study-ready version of LCA to incorporate image viewing and interpretation functionalities, which can be further extended to provide decision support in interpreting the various imaging modalities.

8.2.3. Extending the Task Network Model

One of the major weaknesses in our guideline rule inference framework is that it does not contain temporal concepts to account for clinical workflows and the dynamics of processes as most CIG formalisms do. This is partly due to the LUCADA data model which does not portray the patient journey in a temporal manner, and partly an intentional choice for the task at hand. Within the framework introduced in this thesis, although the data is not kept longitudinally, as treatment or outcome related variables become available, they automatically trigger guideline rules that involve them.

However, similar to EON's non-monolithic Task Network Model, the argumentation domain within our ontological framework can be extended to incorporate classes and properties that would allow the definitions of care plans which incorporate sequential and clinical workflow management. The use of a purely ontological approach allows the extension of the ontology as long as it is within the boundaries of OWL-2.

8.3. Final Comments

"*Medicine is a humanly impossible task*" [296], is one of the most commonly used quotes in the field of Medical Informatics and CDS. We believe that this is an accurate yet incomplete statement. Our first-hand experience throughout this research convinced us that, for the time being, medicine is an impossible task for computers as well. The many intricate details and subtleties in care decisions are not yet manageable by machines alone due to the lack of data to represent them, if for nothing else.

However, as we have showcased in this thesis, computers acting as ever-attentive personal assistants to the clinicians, can assist decision making processes in various ways. The fact that different decision support approaches, such as rule-based, probabilistic or case-based inference, derive from fundamentally different research hypotheses does not render them incompatible. On the contrary, giving different views on the same matter, we believe they fulfil complementary roles in CDS.

The major obstacle in the wider adoption of CDS systems is the lack of comprehensive electronic patient records that contain structured and diverse patient data. In many fields of medicine, patient data are still kept in paper records or in free-text electronic entries [165]. Therefore, benefiting from a CDS platform comes at the expense of having to enter patients manually into the system, which adds to the clinician's workload. In this setting, deprived of comprehensive and easily accessible electronic data, guideline rule-based

decision support still fulfils an important role in representing expert knowledge that takes into consideration multiple criteria. However, we believe that as the amount and breadth of centralised electronic patient data increase, sophisticated probabilistic inference tools, such as Bayesian networks or Decision networks, have a high potential to displace the central role of guideline rule-based decision support.

Bibliography

[1] R. A. Weinberg, "The Nature of Cancer," in in *The Biology of Cancer*, 2006, pp. 25–56.

[2] Cancer Research UK, "What Cancer is," *About Cancer*, 2012. [Online]. Available: http://cancerhelp.cancerresearchuk.org/about-cancer/what-is-cancer/cells/what-cancer-is. [Accessed: 25-Sep-2012].

[3] National Collaborating Centre for Cancer, "The diagnosis and treatment of lung cancer (update)," 2011.

[4] NICE, "Quick Reference Guide Lung Cancer," 2011.

[5] WHO, "Cancer fact sheet No 297," 2012. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs297/en/. [Accessed: 25-Sep-2011].

[6] L. Crinò, W. Weder, J. van Meerbeeck, and E. Felip, "Early stage and locally advanced (non-metastatic) non-small-cell lung cancer: ESMO Clinical Practice Guidelines for diagnosis, treatment and follow-up.," *Ann. Oncol.*, vol. 21 Suppl 5, no. Supplement 5, pp. v103–15, May 2010.

[7] E. Lim, D. Baldwin, M. Beckles, J. Duffy, J. Entwisle, C. Faivre-Finn, K. Kerr, A. Macfie, J. McGuigan, S. Padley, S. Popat, N. Screaton, M. Snee, D. Waller, C. Warburton, and T. Win, "Guidelines on the radical management of patients with lung cancer," Oct. 2010.

[8] G. P. Westert and M. Faber, "Commentary: the Dutch approach to unwarranted medical practice variation," *Br. Med. J.*, p. 342: d1429, 2011.

[9] Cancer Research UK, "Surgery to treat cancer," 2012. [Online]. Available: http://cancerhelp.cancerresearchuk.org/about-cancer/treatment/surgery/surgery-to-treat-cancer. [Accessed: 25-Sep-2012].

[10] K. R. Yabroff, T. S. McNeel, W. R. Waldron, W. W. Davis, M. L. Brown, S. Clauser, and W. F. Lawrence, "Health limitations and quality of life associated with cancer and other chronic diseases by phase of care," *Med. Care*, vol. 45, no. 7, pp. 629–37, Jul. 2007.

[11] C. Schag, P. Ganz, D. Wing, M. Sim, and J. Lee, "Quality of life in adult survivors and prostate cancer of lung , colon," *Qual. Life Res.*, vol. 3, pp. 127–141, 1994.

[12] C. Y. Ko, M. Maggard, and E. H. Livingston, "Evaluating health utility in patients with melanoma, breast cancer, colon cancer, and lung cancer: a nationwide, population-based assessment," *J. Surg. Res.*, vol. 114, no. 1, pp. 1–5, Sep. 2003.

[13] Cancer Research UK, "About Lung Cancer - A Quick Guide," 2012. [Online]. Available: http://cancerhelp.cancerresearchuk.org/type/lung-cancer/.

[14] R. Rami-Porta, J. J. Crowley, and P. Goldstraw, "The revised TNM staging system for lung cancer.," *Ann. Thorac. Cardiovasc. Surg.*, vol. 15, no. 1, pp. 4–9, Feb. 2009.

[15] P. Goldstraw, J. Crowley, K. Chansky, D. J. Giroux, P. a Groome, R. Rami-Porta, P. E. Postmus, V. Rusch, and L. Sobin, "The IASLC Lung Cancer Staging Project: proposals for the revision of the TNM stage groupings in the forthcoming (seventh) edition of the TNM Classification of malignant tumours.," *J. Thorac. Oncol.*, vol. 2, no. 8, pp. 706–14, Aug. 2007.

[16] The National Lung Cancer Audit, "The National Clinical Lung Cancer Audit (LUCADA) Data Manual," 2012.

[17] K. Calman and H. Deirdre, "A policy framework for commissioning cancer services," 1995.

[18] M. Austin, "Information Integration and Decision Support for Multidisciplinary Team Meetings on Colorectal Cancer," Oxford University, 2008.

[19] B. W. Lamb, K. F. Brown, K. Nagpal, C. Vincent, J. S. a Green, and N. Sevdalis, “Quality of care management decisions by multidisciplinary cancer teams: a systematic review.,” *Ann. Surg. Oncol.*, vol. 18, no. 8, pp. 2116–25, Aug. 2011.

[20] National Cancer Action Team, “Multi-disciplinary Team Development,” 2012. [Online]. Available: http://ncat.nhs.uk/our-work/ensuring-better-treatment/multi-disciplinary-team-development#. [Accessed: 26-Sep-2012].

[21] A. Lanceley, J. Savage, U. Menon, and I. Jacobs, “Influences on multidisciplinary team decision-making,” *Int. J. Gynecol. Cancer*, vol. 18, pp. 215–222, 2008.

[22] D. M. Eddy, “The Challenge,” *J. Am. Med. Assoc.*, vol. 263, no. 2, pp. 287–290, 1990.

[23] 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*, vol. 280, no. 15, pp. 1339–46, Oct. 1998.

[24] E. S. Berner, “Clinical Decision Support Systems : State of the Art,” 2009.

[25] R. A. Miller, “Why the standard view is standard: people, not machines, understand patients’ problems,” *J. Med. Philos.*, no. 15, pp. 581–591, 1990.

[26] V. Patkar, D. Acosta, T. Davidson, A. Jones, J. Fox, and M. Keshtgar, “Using computerised decision support to improve compliance of cancer multidisciplinary meetings with evidence-based guidance.,” *BMJ Open*, vol. 2, no. 3, Jan. 2012.

[27] V. Patkar, C. Hurt, R. Steele, S. Love, a Purushotham, M. Williams, R. Thomson, and J. Fox, “Evidence-based guidelines and decision support services: A discussion and evaluation in triple assessment of suspected breast cancer.,” *Br. J. Cancer*, vol. 95, no. 11, pp. 1490–6, Dec. 2006.

[28] B. Séroussi, J. Bouaud, J. Gligorov, and S. Uzan, “Supporting multidisciplinary staff meetings for guideline-based breast cancer management: a study with OncoDoc2.,” *AMIA Annu. Symp. Proc.*, pp. 656–60, Jan. 2007.

[29] J. Bouaud and B. Séroussi, “Revisiting the EBM decision model to formalize non-compliance with computerized CPGs: results in the management of breast cancer with OncoDoc2.,” *AMIA Annu. Symp. Proc.*, vol. 2011, pp. 125–34, Jan. 2011.

[30] B. W. Lamb, N. Sevdalis, C. Vincent, and J. S. a Green, “Development and evaluation of a checklist to support decision making in cancer multidisciplinary team meetings: MDT-QuIC.,” *Ann. Surg. Oncol.*, vol. 19, no. 6, pp. 1759–65, Jun. 2012.

[31] A. Gawande, “The Checklist,” *The New Yorker*, 2007.

[32] J. Fox, N. Johns, C. Lyons, A. Rahmanzadeh, R. Thomson, and P. Wilson, “PROforma: a general technology for clinical decision support systems.,” *Comput. Methods Programs Biomed.*, vol. 54, no. 1–2, pp. 59–67, 1997.

[33] R. A. Miller, “Medical Diagnostic Decision Support Systems -Past , Present , and Future : a threaded bibliographyand brief commentary,” *J. Am. Med. Informatics Assoc.*, vol. 1, pp. 8–27, 1994.

[34] J. A. Osheroff, *Improving Medication Use and Outcomes With Clinical Decision Support: Step-By-Step Guide*, 1st ed. U.S.: United States Pharmacopeial, 2009.

[35] K. Kawamoto, C. a Houlihan, E. A. Balas, and D. F. Lobach, “Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success.,” *BMJ*, vol. 330, no. 7494, p. 765, Apr. 2005.

[36] M. B. Şeşen, R. Banares-Alcantara, J. Fox, T. Kadir, and M. Brady, “Lung Cancer Assistant : An Ontology-Driven , Online Decision Support Prototype for Lung Cancer Treatment Selection,” in *OWL Experiences and Directions*, 2012.

[37] M. B. Şeşen, T. Kadir, R. Banares-Alcantara, J. Fox, and M. Brady, "Survival Prediction and Treatment Recommendation with Bayesian Techniques in Lung Cancer," in *American Medical Informatics Association*, 2012, pp. 838–848.

[38] D. Isern and A. Moreno, "Computer-based execution of clinical guidelines: a review.," *Int. J. Med. Inform.*, vol. 77, no. 12, pp. 787–808, Dec. 2008.

[39] P. D. E. Clercq, K. Kaiser, and A. Hasman, "Computer-interpretable Guideline Formalisms," *Stud. Health Technol. Inform.*, vol. 139, pp. 22–43, 2008.

[40] A. Garg, N. K. J. Adhikari, J. Beyene, J. Sam, and R. B. Haynes, "Effects of Computerized Clinical Decision Support Systems on Practitioner Performance," *J. Am. Med. Assoc.*, vol. 293, no. 10, pp. 1223–1238, 2005.

[41] M. D. Cabana, "Why Don't Physicians Follow Clinical Practice Guidelines?: A Framework for Improvement," *JAMA J. Am. Med. Assoc.*, vol. 282, no. 15, pp. 1458–1465, Oct. 1999.

[42] M. Peleg, O. Ogunyemi, S. Tu, A. Boxwala, Q. Zeng, R. Greenes, and E. Shortliffe, "Using features of Arden Syntax with object-oriented medical data models for guideline modeling.," *Proc. AMIA Symp.*, pp. 523–7, Jan. 2001.

[43] G. Jamtvedt, J. Young, D. Kristoffersen, M. OBrien, and A. Oxman, "Audit and feedback : effects on professional practice and health care outcomes (Review)," *Cochrane Rev.*, no. 4, 2003.

[44] K. Dube, "A generic approach to supporting the management of computerised clinical guidelines and protocols," 2004.

[45] P. Ciccarese, E. Caffie, L. Boiocchi, S. Quaglini, and S. M, "A guideline management system," *Stud. Health Technol. Inform.*, vol. 107, 2004.

[46] P. D. Johnson, S. Tu, N. Booth, B. Sugden, and I. N. Purves, "Using scenarios in chronic disease management guidelines for primary care.," *Proc. AMIA Symp.*, pp. 389–393, 2000.

[47] P. Terenziani, L. Anselma, A. Bottrighi, L. Giordano, and S. Montani, "Automatic checking of the correctness of clinical guidelines in GLARE.," *Stud. Health Technol. Inform.*, vol. 129, no. Pt 1, pp. 807–811, 2007.

[48] M. Peleg, S. Tu, and J. Bury, "Comparing Guideline Models : A Case-study Approach," *J. Am. Med. Informatics Assoc.*, pp. 52–68, 2003.

[49] D. Sutton and J. Fox, "The Syntax and Semantics of the PRO forma Guideline Modeling Language," *J. Am. Med. Informatics Assoc.*, vol. 10, no. 5, pp. 433–443, 2003.

[50] M. Sordo, J. Fox, C. Blum, P. Taylor, R. Lee, and E. Alberdi, "Combining decision support and image processing: a PROforma model.," *Stud. Health Technol. Inform.*, vol. 84, no. Pt 1, pp. 547–51, Jan. 2001.

[51] COSSAC, "Knowledge Modelling," 2010. [Online]. Available: http://www.cossac.org/technologies/proforma/modelling. [Accessed: 12-Aug-2010].

[52] S. Miksch, Y. Shahar, and P. Johnson, "ASBRU: A task Specific, Intention-based, and Time Oriented Language for Representing Skeletal Plans," *Aids*, pp. 1–25, 1997.

[53] Y. Shahar, O. Young, E. Shalom, M. Galperin, A. Mayaffit, R. Moskovitch, and A. Hessing, "A framework for a distributed, hybrid, multiple-ontology clinical-guideline library, and automated guideline-support tools.," *J. Biomed. Inform.*, vol. 37, no. 5, pp. 325–344, 2004.

[54] "Protocure II," 2006. [Online]. Available: http://www.protocure.org/. [Accessed: 18-Nov-2011].

[55] "MobiGuide," 2012. [Online]. Available: http://www.mobiguide-project.eu/. [Accessed: 18-Dec-2012].

[56] A. Boxwala, M. Peleg, S. Tu, O. Ogunyemi, Q. T. Zeng, D. Wang, V. L. Patel, R. a Greenes, and E. H. Shortliffe, "GLIF3: a representation format for sharable computer-interpretable clinical practice guidelines.," *J. Biomed. Inform.*, vol. 37, no. 3, pp. 147–61, Jun. 2004.

[57] M. Sordo, O. Ogunyemi, A. Boxwala, R. A. Greenes, and S. Tu, "GELLO: A Common Expression Language," *Harvard University*, 2005. [Online]. Available: http://www.dsg.harvard.edu/uploads/Margarita/GELLO-073105short.htm. [Accessed: 17-Dec-2012].

[58] D. Wang, M. Peleg, S. Tu, A. a Boxwala, O. Ogunyemi, Q. Zeng, R. a Greenes, V. L. Patel, and E. H. Shortliffe, "Design and implementation of the GLIF3 guideline execution engine.," *J. Biomed. Inform.*, vol. 37, no. 5, pp. 305–18, Oct. 2004.

[59] X. H. Le, A. Luque, D. Wang, and R. Medical, "Development of Guideline-Driven Mobile Applications for Clinical Education and Decision Support with Customization to Individual Patient Cases," in *American Medical Informatics Association*, 2012, vol. 37, no. 3.

[60] M. Musen, S. Tu, and K. Das, "EON: a component-based approach to automation of protocol-directed therapy," *J. Am. Med. Informatics Assoc.*, vol. 3, no. 6, 1996.

[61] S. Tu and J. Glasgow, "SAGE Guideline Model Technical Specification," *October*, 2006.

[62] D. Berg, P. Ram, J. Glasgow, and J. Castro, "SAGEDesktop: an environment for testing clinical practice guidelines.," *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, vol. 5, pp. 3217–20, Jan. 2004.

[63] P. Ram, D. Berg, S. Tu, G. Mansfield, Q. Ye, R. Abarbanel, and N. Beard, "Executing clinical practice guidelines using the SAGE execution engine.," *Stud. Health Technol. Inform.*, vol. 107, no. Pt 1, pp. 251–5, Jan. 2004.

[64] S. Tu, J. R. Campbell, J. Glasgow, M. Nyman, R. McClure, J. McClay, C. Parker, K. M. Hrabak, D. Berg, T. Weida, J. Mansfield, M. Musen, and R. M. Abarbanel, "The SAGE Guideline Model: achievements and overview.," *J. Am. Med. Inform. Assoc.*, vol. 14, no. 5, pp. 589–98, 2006.

[65] D. Isern, D. S, and A. Moreno, "HeCaSe2 : A Multi-agent Ontology-Driven Guideline Enactment Engine," *IEEE Intell. Syst.*, pp. 322–324, 2007.

[66] K. Kawamoto, "Open CDS," 2012. [Online]. Available: http://www.opencds.org/. [Accessed: 10-Oct-2012].

[67] T. Gruber, L. Liu, and T. Ozsu, "Ontology," *Encyclopedia of Database Systems*. 2009.

[68] M. B. Şeşen, P. Suresh, R. Banares-Alcantara, and V. Venkatasubramanian, "An ontological framework for automated regulatory compliance in pharmaceutical manufacturing," *Comput. Chem. Eng.*, vol. 34, no. 7, pp. 1155–1169, Jul. 2010.

[69] I. Horrocks, "An Introduction to Description Logics." 2001.

[70] A. S. Bechhofer, C. Goble, I. Horrocks, and E. S. Bechhofer, "Requirements of Ontology Languages," 2002.

[71] M. Horridge, H. Knublauch, A. Rector, R. Stevens, C. Wroe, S. Jupp, G. Moulton, and N. Drummond, "A Practical Guide To Building OWL Ontologies Using Protege 4 and CO-ODE Tools Edition 1.2," 2009.

[72] M. B. Şeşen, "An Ontology-Driven Decision Support Prototype for Breast Cancer Chemotherapy Planning," 2010.

[73] O. Lassila and D. Mcguinness, "The Role of Frame-Based Representation on the Semantic Web Introduction : Frame-based Representation Systems," 1992.

[74] I. Horrocks, D. Fensel, J. Broekstra, S. Decker, M. Erdmann, C. Goble, F. Van Harmelen, M. Klein, S. Staab, R. Studer, E. Motta, and V. U. Amsterdam, "The Ontology Inference Layer OIL," 2000.

[75] W3C, "The DARPA Agent Markup Language Homepage," *2006*. [Online]. Available: www.daml.org. [Accessed: 09-Oct-2010].

[76] W3C, "Resource Description Framework (RDF)," 2004. [Online]. Available: http://www.w3.org/RDF/.

[77] W3C, "DAML+OIL (March 2001) Reference Description," 2001.

[78] W3C, "OWL Web Ontology Language Overview," 2004. [Online]. Available: http://www.w3.org/TR/owl-features/.

[79] E. Jimenez-Ruiz, B. C. Grau, Y. Zhou, and I. Horrocks, "Large-scale Interactive Ontology Matching : Algorithms and Implementation," in *European Conference on Artificial Intelligence*, 2012, no. ii, pp. 0–5.

[80] World Wide Web Consortium, "OWL 2," 2009. [Online]. Available: http://www.w3.org/TR/2009/PR-owl2-overview-20090922/.

[81] W3C, "OWL 2 Web Ontology Language Manchester Syntax (Second Edition)," 2012. [Online]. Available: http://www.w3.org/TR/2012/NOTE-owl2-manchester-syntax-20121211/.

[82] D. Tsarkov and I. Horrocks, "FaCT ++ Description Logic Reasoner : System Description," *System*, pp. 292–297, 2006.

[83] Stanford Center for Biomedical Informatics Research, "Protege 4.1," 2011. [Online]. Available: http://protege.stanford.edu/.

[84] W3C, "OWL 2 Web Ontology Language New Features and Rationale," 2009. [Online]. Available: http://www.w3.org/TR/2009/REC-owl2-new-features-20091027/. [Accessed: 01-Oct-2011].

[85] M. Horridge and S. Bechhofer, "OWL API Version 3.2.3," 2011. [Online]. Available: http://owlapi.sourceforge.net/.

[86] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, "Pellet: A practical OWL-DL reasoner," *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 5, no. 2, pp. 51–53, Jun. 2007.

[87] A. Kershenbaum, A. Fokoue, C. Patel, C. Welty, J. Cimino, L. Ma, K. Srinivas, R. Schloss, and J. William, "A View of OWL from the Field : Use cases and Experiences," *Text*, vol. 216, pp. 3–6, 2006.

[88] W3C, "OWL 2 Implementations," 2012. [Online]. Available: http://www.w3.org/2007/OWL/wiki/Implementations.

[89] Information Systems Group, "HermiT OWL Reasoner," *Information Systems Group*, 2013. [Online]. Available: http://hermit-reasoner.com/. [Accessed: 25-Feb-2013].

[90] B. Glimm, I. Horrocks, B. Motik, and G. Stoilos, "Optimising Ontology Classification," in *International Semantic Web Conference*, 2010, pp. 225–240.

[91] Clark & Parsia, "Pellet: OWL 2 Reasoner for Java," 2011. [Online]. Available: http://clarkparsia.com/pellet/. [Accessed: 11-Nov-2011].

[92] Racer Systems Gmbh Co, "RacerPro 2.0," 2012. [Online]. Available: http://www.racer-systems.com/index.phtml. [Accessed: 06-Jan-2012].

[93] B. Suntisrivaraporn, "Module Extraction and Incremental Classiffication: A Pragmatic Approach for EL+ Ontologies," 2007.

[94] J. Mendez, “JCel,” 2010. [Online]. Available: http://jcel.sourceforge.net/index.html. [Accessed: 07-Jan-2012].

[95] Y. Kazakov, M. Krotzsch, and F. Simancik, “Concurrent Classification of EL Ontologies Technical Report,” 2011.

[96] M. J. Lawley, “Fast Classification in Protege : Snorocket as an OWL 2 EL Reasoner,” *Adv. Ontol.*, vol. 122, 2010.

[97] F. Baader, I. Horrocks, and U. Sattler, “Description Logics,” in in *Handbook of Knowledge Representation*, 2007, pp. 135–180.

[98] N. Sioutos, S. de Coronado, M. W. Haber, F. W. Hartel, W.-L. Shaiu, and L. W. Wright, “NCI Thesaurus: a semantic model integrating cancer-related clinical and molecular information.,” *J. Biomed. Inform.*, vol. 40, no. 1, pp. 30–43, Feb. 2007.

[99] U.S National Library of Medicine, “UMLS,” 2011. [Online]. Available: http://www.nlm.nih.gov/research/umls/.

[100] C. Rosse and J. L. V. Mejino, “The Foundational Model of Anatomy Ontology,” *Comput. Biol.*, vol. 6, pp. 59–117, 2008.

[101] National Cancer Institute, “NCI Thesaurus,” 2011. [Online]. Available: http://nciterms.nci.nih.gov/ncitbrowser/pages/multiple_search.jsf;jsessionid=24E63B0DC6206FE9849DDF48366FF25C.

[102] IHTSDO, “About SNOMED CT.” [Online]. Available: http://www.ihtsdo.org/snomed-ct/snomed-ct0/. [Accessed: 20-Aug-2010].

[103] IHTSDO, “SNOMED Clinical Terms User Guide July 2010 International Release (US English),” 2010.

[104] The National Center for Biomedical Ontology, “BioPortal,” 2013. [Online]. Available: http://bioportal.bioontology.org/. [Accessed: 01-Jul-2012].

[105] U.S. National Library of Medicine, “The UMLS Source Vocabularies,” 2011. [Online]. Available: http://www.nlm.nih.gov/research/umls/knowledge_sources/metathesaurus/release/source_vocabularies.html.

[106] National Center for Biotechnology Information, “The UMLS Reference Manual,” 2011. [Online]. Available: http://www.ncbi.nlm.nih.gov/books/NBK9675/#ch01.I11_Purpose_of_the_UMLS.

[107] A. Kumar and B. Smith, “Oncology Ontology in the NCI Thesaurus,” in *AIME 2005*, 2005, pp. 213 – 220.

[108] NHS Connecting for Health, “Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT).” [Online]. Available: http://www.connectingforhealth.nhs.uk/systemsandservices/etd/elearning/nhselearning/snomedct/index_html/?searchterm=snomed. [Accessed: 21-Aug-2010].

[109] M. Caminada and L. Amgoud, “On the evaluation of argumentation formalisms,” *Artif. Intell.*, vol. 171, pp. 286–310, 2007.

[110] J. Fox, D. Glasspool, D. Grecu, S. Modgil, and M. South, “Argumentation-based Inference and Decision Making — A Medical Perspective,” *IEEE Intell. Syst.*, vol. 22, pp. 34–41, 2007.

[111] N. Gorogiannis, A. Hunter, and M. Williams, “An Argument-based Approach to Reasoning with Clinical Knowledge,” *Int. J. Approx. Reason.*, vol. Volume: 51, no. June, pp. 1–22, 2009.

[112] M. Williams and J. Williamson, “Combining Argumentation and Bayesian Nets for Breast Cancer Prognosis,” *J. Logic, Lang. Inf.*, vol. 15, no. 1–2, pp. 155–178, Jul. 2006.

[113] J. Kohlas, "Probabilistic argumentation systems A new way to combine logic with probability," *J. Appl. Log.*, vol. 1, no. 3–4, pp. 225–253, Jun. 2003.

[114] P. J. Lucas, L. van der Gaag, and A. Abu-Hanna, "Bayesian networks in biomedicine and health-care.," *Artif. Intell. Med.*, vol. 30, no. 3, pp. 201–14, Mar. 2004.

[115] K. Korb and A. Nicholson, *Bayesian Artificial Intelligence*, 2nd ed., vol. 52, no. 2. 2010.

[116] R. Daly, Q. Shen, and S. Aitken, "Learning Bayesian networks: approaches and issues," *Knowl. Eng. Rev.*, vol. 26, no. 02, pp. 99–157, May 2011.

[117] D. Heckerman, "A Tutorial on Learning with Bayesian Networks," Redmond, 1996.

[118] K. Murphy, "The Bayes Net Toolbox for MatLab," 2001.

[119] C. Huang, "Inference in belief networks: A procedural guide," *Int. J. Approx. Reason.*, vol. 15, no. 3, pp. 225–263, Oct. 1996.

[120] J. Pearl and S. Russell, "Bayesian Networks," 2000.

[121] D. Barber, *Bayesian Reasoning and Machine Learning*. 2011.

[122] M. J. Flores, A. Nicholson, A. Brunskill, K. Korb, and S. Mascaro, "Incorporating expert knowledge when learning Bayesian network structure: a medical case study.," *Artif. Intell. Med.*, vol. 53, no. 3, pp. 181–204, Nov. 2011.

[123] A. Dekker, C. Dehing-Oberije, D. De Ruysscher, P. Lambin, K. Komati, G. Fung, S. Yu, A. Hope, W. De Neve, and Y. Lievens, "Survival Prediction in Lung Cancer Treated with Radiotherapy: Bayesian Networks vs. Support Vector Machines in Handling Missing Data," *Int. Conf. Mach. Learn. Appl.*, pp. 494–497, Dec. 2009.

[124] K. Jayasurya, G. Fung, S. Yu, C. Dehing-Oberije, D. De Ruysscher, a. Hope, W. De Neve, Y. Lievens, P. Lambin, and a. L. a. J. Dekker, "Comparison of Bayesian network and support vector machine models for two-year survival prediction in lung cancer patients treated with radiotherapy," *Med. Phys.*, vol. 37, no. 4, p. 1401, 2010.

[125] I. Beinlich, H. Suermondt, R. Chavez, and C. G., "The ALARM monitoring system," in *European Conference on Artificial Intelligence in medicine*, 1992, pp. 689–699.

[126] A. Onisko, M. Druzdzel, and H. Wasyluk, "A Probabilistic Causal Model for Diagnosis of Liver Disorders," in *Symposium on Intelligent Information Systems*, 1998, pp. 379–387.

[127] L. van der Gaag, S. Renooij, C. L. Witteman, B. M. Aleman, and B. G. Taal, "Probabilities for a probabilistic network: a case study in oesophageal cancer.," *Artif. Intell. Med.*, vol. 25, no. 2, pp. 123–48, Jun. 2002.

[128] D. Rubin, E. Burnside, and R. Shachter, "A Bayesian Network to assist mammography interpretation," *Oper. Res. Heal. Care*, pp. 695–720, 2005.

[129] C. R. Twardy, A. E. Nicholson, and K. Korb, "Knowledge engineering cardiovascular Bayesian networks from the literature," 2005.

[130] "Promedas," 2013. [Online]. Available: http://www.promedas.nl/online/.

[131] T. Boneh, "Ontology and Bayesian Decision Networks for Supporting the Meteorological Forecasting Process," Monash University, 2010.

[132] R. W. Robinson, "Counting unlabeled acyclic digraphs," *Comb. Math. V*, vol. 622, pp. 28–43, 1977.

[133] P. Leray and O. Franc, "Bayesian Network Structure Learning and Incomplete Data," in *International and Indeterdisciplinary Conference on Adaptive Knowledge Representation and Reasoning*, 2005, pp. 33–40.

[134] D. M. Chickering, "Large-Sample Learning of Bayesian Networks is NP-Hard," *J. Mach. Learn. Res.*, vol. 5, pp. 1287–1330, 2004.

[135] W. Buntine, “A guide to the literature on learning probabilistic networks from data,” *IEEE Trans. Knowl. Data Eng.*, 1996.

[136] K. Murphy, “Learning Bayes net Structure from sparse data sets,” 2001.

[137] T. Verma and J. Pearl, “Equivalence and Synthesis of Causal Models,” in *Uncertainty in Artificial Intelligence*, 1990, pp. 220–227.

[138] P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search*, 2nd ed. MIT Press, 2000.

[139] A. M. Carvalho, “Scoring functions for learning Bayesian networks.” pp. 1–48, 2012.

[140] C. Wallace and D. M. Boulton, “An Information Measure for Classification,” *Comput. J.*, vol. 11, no. 2, pp. 185–194, 1968.

[141] J. Rissanen, “Modeling by shortest data description,” *Automatica*, vol. 14, pp. 465–471, 1978.

[142] K. Murphy, “Active Learning of Causal Bayes net Structure,” 2001.

[143] G. F. Cooper and E. Herskovits, “A Bayesian Method for the Induction of Probabilistic Networks from Data,” vol. 347, pp. 309–347, 1992.

[144] J. Neil and K. Korb, “The Evolution of Causal Models: A Comparison of Bayesian Metrics and Structure Priors,” in *Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining*, 1999, pp. 432–437.

[145] S. Andreassen, M. Woldbye, and B. Falck, “MUNIN - A Causal Probabilistic Network for Interpretation of Electromyographic Findings*,” *Knowl. Represent.*, pp. 366–372, 1986.

[146] S. Andreassen, Ch. Riekehr, B. Kristensen, H. Schonheyder, and L. Leibovici, “Using probabilistic and decision-theoretic methods in treatment and prognosis modelling,” *Artif. Intell. Med.*, vol. 15, pp. 121–134, 1999.

[147] S. F. Galán, F. Aguado, F. J. Díez, and J. Mira, “NasoNet , Joining Bayesian Networks and Time to Model Nasopharyngeal Cancer Spread,” *Artif. Intell. Med.*, no. July, pp. 207–216, 2001.

[148] D. Heckerman, E. Horvitz, and B. Nathwani, “Toward Normative Expert Systems: The Pathfinder Project,” *Meth Inf. Med*, 1992.

[149] E. M. Helsper and L. C. Van Der Gaag, “Building Bayesian Networks through Ontologies,” in *ECAI 2002*, 2002.

[150] P. J. Lucas, N. C. de Bruijn, K. Schurink, and a Hoepelman, “A probabilistic and decision-theoretic approach to the management of infectious disease at the ICU.,” *Artif. Intell. Med.*, vol. 19, no. 3, pp. 251–79, Jul. 2000.

[151] P. J. Lucas, H. Boot, and B. G. Taal, “Computer-based decision support in the management of primary gastric non-Hodgkin lymphoma.,” *Methods Inf. Med.*, vol. 37, no. 3, pp. 206–19, Sep. 1998.

[152] C. Wallace and K. Korb, “Causal Discovery via MML,” *Causal Model. Intell. Data Manag.*, pp. 89–111, 1999.

[153] I. J. Myung, “Tutorial on maximum likelihood estimation,” *J. Math. Psychol.*, vol. 47, no. 1, pp. 90–100, Feb. 2003.

[154] A. Eshky, “Bayesian Methods of Parameter Estimation,” 2009.

[155] M. A. Chappell, A. Groves, and M. W. Woolrich, “The FMRIB Variational Bayes Tutorial : Variational Bayesian inference for a non-linear forward model,” Oxford, 2007.

[156] A. Kak, “ML , MAP , and Bayesian — The Holy Trinity of Parameter Estimation and Data Prediction,” 2010.

[157] J. Pearl, “Reverend Bayes on Inference Engines: A Distributed Hierarchical Approach,” in *AAAI-82*, 1982, pp. 133–136.

[158] J. Kim and J. Pearl, “A Computational Model for Causal and Diagnostic Reasoning in Inference Systems,” 1983.

[159] S. Lauritzen and D. Spiegelhalter, “Local Computations with Probabilities on Graphical Structures and their Application to Expert Systems,” *J. R. Stat. Soc.*, vol. 50, no. 2, pp. 157–224, 1988.

[160] R. D. Shachter, “Intelligent Probabilistic Inference,” *Uncertain. Artif. Intell.*, pp. 371–382, 1986.

[161] The National Lung Cancer Audit, “Clinical Audit Support Programme - Lung Cancer,” 2012. [Online]. Available: http://www.ic.nhs.uk/services/national-clinical-audit-support-programme-ncasp/cancer/lung.

[162] WHO, “International Statistical Classification of Diseases and Related Health Problems 10th Revision,” 2010. [Online]. Available: http://apps.who.int/classifications/apps/icd/icd10online/.

[163] M. Coleman, D. Forman, H. Bryant, J. Butler, and M. Richards, “Cancer survival in Australia, Canada, Denmark, Norway, Sweden, and the UK, 1995-2007 (the International Cancer Benchmarking Partnership): an analysis of population-based cancer registry data.,” *Lancet*, vol. 377, no. 9760, pp. 127–38, Jan. 2011.

[164] L. Holmberg, F. Sandin, F. Bray, M. Richards, J. Spicer, M. Lambe, A. Klint, M. Peake, T.-E. Strand, K. Linklater, D. Robinson, and H. Møller, “National comparisons of lung cancer survival in England, Norway and Sweden 2001-2004: differences occur early in follow-up.,” *Thorax*, vol. 65, no. 5, pp. 436–41, May 2010.

[165] T. Benson, *Principles of Health Interoperability HL7 and SNOMED*. 2010.

[166] B. C. Grau, I. Horrocks, and U. Sattler, “Modular Reuse of Ontologies : Theory and Practice,” *J. Artif. Intell. Res.*, vol. 31, pp. 273–318, 2008.

[167] The PostgreSQL Global Development Group, “PostgreSQL JDBC Driver.” 2012.

[168] N. Konstantinou, D.-E. Spanos, and N. Mitrou, “Ontology and database mapping: a survey of current implementations and future directions,” *J. Web Eng.*, vol. 7, no. 1, pp. 1–24, Mar. 2008.

[169] N. F. Noy and C. D. Hafner, “The State of the Art in Ontology Design: A Survey and Comparative Review,” *Adv. Artif. Intell.*, vol. 18, no. 3, pp. 53–74, 1997.

[170] I. Astrova and B. Stantic, “Reverse Engineering of Relational Databases to Ontologies : An Approach Based on an Analysis of HTML Forms,” in *European Semantic Web Symposium*, 2004, pp. 327–341.

[171] J. Euzenat, C. Meilicke, H. Stuckenschmidt, P. Shvaiko, and C. Trojahn, “Ontology Alignment Evaluation Initiative : Six Years of Experience,” *J. Data Semant.*, pp. 158–192, 2011.

[172] P. Shvaiko, “Ontology Matching: State of the Art and Future Challenges,” *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 1, pp. 158–176, Jan. 2011.

[173] Y. R. Jean-Mary, E. P. Shironoshita, and M. R. Kabuka, “Ontology Matching with Semantic Verification,” *Web Semant.*, vol. 7, no. 3, pp. 235–251, 2009.

[174] T. Kirsten, A. Gross, M. Hartung, and E. Rahm, “GOMMA: a component-based infrastructure for managing and analyzing life science ontologies and their evolution.,” *J. Biomed. Semantics*, vol. 2, no. 1, p. 6, Jan. 2011.

[175] E. Jimenez-Ruiz and B. C. Grau, “LogMap : Logic-based and Scalable Ontology Matching,” in *International Semantic Web Conference*, 2011.

[176] P. Lambrix and H. Tan, “SAMBO - A System for Aligning and Merging,” *J. ofWeb Semant.*, vol. 4, no. 3, pp. 196–206, 2006.

[177] W. Hu, Y. Qu, and G. Cheng, “Matching large ontologies: A divide-and-conquer approach,” *Data Knowl. Eng.*, vol. 67, no. 1, pp. 140–160, Oct. 2008.

[178] Q. Reul and J. Z. Pan, “KOSIMap : Use of Description Logic Reasoning to Align Heterogeneous Ontologies,” in *Workshop on Description Logics*, 2010, pp. 497–508.

[179] E. Jimenez-Ruiz, B. C. Grau, and I. Horrocks, “LogMap : Scalable , Logic-based and Interactive Ontology Matching,” in *International Semantic Web COnference*, 2011, pp. 273–288.

[180] International Health Terminology Standards Development Organisation, “SNOMED CT ® Technical Implementation Guide,” 2013.

[181] J. Seidenberg and A. Rector, “Web Ontology Segmentation : Analysis , Classification and Use,” in *iIternational conference on World Wide Web*, 2006, pp. 13–22.

[182] C. Lutz, D. Walther, and F. Wolter, “Conservative Extensions in Expressive Description Logics,” in *The International Joint Conference on Artificial Intelligence (IJCAI-07),*, 2007, no. i, pp. 453–459.

[183] B. C. Grau, I. Horrocks, Y. Kazakov, and U. Sattler, “Just the Right Amount : Extracting Modules from Ontologies ∗,” in *16th International Confe- rence on World Wide Web (WWW-2007)*, 2007, pp. 717–726.

[184] E. Jimenez-Ruiz, U. Sattler, and T. Schneider, “Locality Module Extractor.” 2007.

[185] “CliniClue Explore,” 2011. [Online]. Available: http://www.cliniclue.com/.

[186] E. Jimenez-Ruiz, B. C. Grau, U. Sattler, T. Schneider, and R. Berlanga, “Safe and Economic Re-Use of Ontologies : A Logic-Based Methodology and Tool Support,” *Online*, vol. 5021, no. November, pp. 185–199, 2008.

[187] B. Hu, S. Dasmahapatra, D. Dupplaw, P. Lewis, and N. Shadbolt, “Reflections on a medical ontology,” *Int. J. Hum. Comput. Stud.*, vol. 65, no. 7, pp. 569–582, Jul. 2007.

[188] A. Abu-Hanna, R. Cornet, N. de Keizer, M. Crubézy, and S. Tu, “Protégé As a Vehicle for Developing Medical Terminological Systems,” *Int. J. Hum. Comput. Stud.*, vol. 62, no. 5, pp. 639–663, May 2005.

[189] A. Gomez-Perez, N. Juristo, and J. Pazos, “Evlauation and Assessment of Knowledge Sharing Technology,” in in *Towards Very Large Knowledge Bases*, 1995, pp. 289–296.

[190] I. Horrocks, P. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, “SWRL: A Semantic Web Rule Language Combining OWL and RuleML,” 2004.

[191] T. Strang and C. Linnhoff-Popien, “A Context Modeling Survey,” in *International Workshop on Advanced Context Modelling, Reasoning and Management*, 2004.

[192] V. Kashyap, A. Morales, and T. Hongsermeier, “On Implementing Clinical Decision Support : Achieving Scalability and Maintainability by Combining Business Rules and Ontologies,” in *American Medical Informatics Association*, 2006, no. 1, pp. 414–418.

[193] W3C, “SPARQL Query Language for RDF,” 2008. [Online]. Available: http://www.w3.org/TR/rdf-sparql-query/. [Accessed: 01-Aug-2010].

[194] M. H. Williams, “Integrating Ontologies and Argumentation for decision-making in breast cancer,” University College London, 2008.

[195] D. Beimel and M. Peleg, “Using OWL and SWRL to represent and reason with situation-based access control policies,” *Data Knowl. Eng.*, vol. 70, no. 6, pp. 596–615, Jun. 2011.

[196] B. Motik, U. Sattler, and R. Studer, “Query Answering for OWL-DL with rules,” *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 3, no. 1, pp. 41–60, 2005.

[197] G. Bucci, V. Sandrucci, and E. Vicario, “Ontologies and Bayesian Networks in Medical Diagnosis,” *Sci. York*, pp. 1–8, 2011.

[198] I. Watson and F. Marir, “Case-based Reasoning: A review,” *Knowl. Eng. Rev.*, vol. 9, pp. 327–361, 1994.

[199] C. Martinez-Cruz, I. J. Blanco, and M. A. Vila, “Ontologies versus relational databases: are they so different? A comparison,” *Artif. Intell. Rev.*, vol. 38, no. 4, pp. 271–290, Jun. 2011.

[200] T. Dillon, E. Chang, M. Hadzic, and P. Wongthongtham, “Differentiating Conceptual Modelling from Data Modelling, Knowledge Modelling and Ontology Modelling and a Notation for Ontology Modelling,” in *Asia-Pacific Conference on Conceptual Modelling*, 2008, pp. 7–17.

[201] P. Haase, Q. Ji, and R. Volz, “Benchmarking OWL Reasoners,” in *ARea Workshop*, 2008.

[202] Q. Elhaik, M. Rousset, and B. Ycart, “Generating Random Benchmarks for Description Logics,” in *Desciption Logics*, 1998, no. Dl.

[203] P. F. Patel-Schneider and R. Sebastiani, “A New General Method to Generate Random Modal Formulae for Testing Decision Procedures,” *J. Artif. Intell. Res.*, vol. 18, pp. 351–389, 2003.

[204] Z. Pan, “Benchmarking DL Reasoners Using Realistic Ontologies,” in *OWL Experiences and Directions*, 2005.

[205] T. Weithöner, T. Liebig, M. Luther, S. Böhm, F. Von Henke, and O. Noppens, “Real-World Reasoning with OWL,” *Semant. Web Res. Appl.*, vol. 4519, pp. 296–310, 2007.

[206] W3C, “OWL 2 Web Ontology Language Profiles,” 2012. .

[207] E. Jimenez-Ruiz, B. C. Grau, and I. Horrocks, “On the Feasibility of Using OWL 2 DL Reasoners for Ontology Matching Problems,” in *On the Feasibility of Using OWL 2 DL Reasoners for Ontology Matching Problems*, 2012.

[208] Y. Kazakov, M. Krötzsch, and F. Simančík, “ELK Reasoner : Architecture and Evaluation,” in *1st International Workshop on OWL Reasoner Evaluation*, 2012.

[209] B. C. Grau, C. Halaschek-wiener, and Y. Kazakov, “History Matters : Incremental Ontology Reasoning Using Modules,” *Lect. Notes Comput. Sci.*, vol. 4825, pp. 183–196, 2007.

[210] B. Parsia, C. Halaschek-wiener, and E. Sirin, “Towards Incremental Reasoning Through Updates in OWL-DL,” in *Reasoning on the Web Workshop*, 2006.

[211] A. Borgida, M. Lenzerini, and R. Rosati, “Description Logics for Data Bases,” in in *The Description Logic Handbook*, 2nd ed., 2007, pp. 472–495.

[212] R. Stevens, M. Eganaaranguren, K. Wolstencroft, U. Sattler, N. Drummond, M. Horridge, and A. Rector, “Using OWL to model biological knowledge,” *Int. J. Hum. Comput. Stud.*, vol. 65, no. 7, pp. 583–594, 2007.

[213] M. Peleg and S. Tu, *Design patterns for clinical guidelines.*, vol. 47, no. 1. 2009, pp. 1–24.

[214] E. Shalom, Y. Shahar, M. Taieb-Maimon, G. Bar, A. Yarkoni, O. Young, S. B. Martins, L. Vaszar, M. K. Goldstein, Y. Liel, A. Leibowitz, T. Marom, and E. Lunenfeld, “A quantitative assessment of a methodology for collaborative specification and evaluation of clinical guidelines.,” *J. Biomed. Inform.*, vol. 41, no. 6, pp. 889–903, Dec. 2008.

[215] D. S. Ettinger and et.al, “Non-Small Cell Lung Cancer,” 2012.

[216] R. Grol, J. Dalhuijsen, S. Thomas, G. Rutten, and H. Mokkink, “General practice guidelines in general practice : observational study,” *BMJ*, vol. 26, no. 317, pp. 858–861, 1998.

[217] J. S. Burgers, R. P. T. M. Grol, J. O. M. Zaat, T. H. Spies, A. K. van der Bij, and H. G. a Mokkink, “Characteristics of effective clinical guidelines for general practice.,” *Br. J. Gen. Pract.*, vol. 53, no. 486, pp. 15–9, Jan. 2003.

[218] S. Codish and R. N. Shiffman, “A model of ambiguity and vagueness in clinical practice guideline recommendations.,” *AMIA Annu. Symp. Proc.*, pp. 146–50, Jan. 2005.

[219] Google, “Google Web Toolkit version 2.4,” 2011. [Online]. Available: http://code.google.com/webtoolkit/.

[220] Google, “Making Remote Procedure Calls,” 2012. [Online]. Available: https://developers.google.com/web-toolkit/doc/latest/tutorial/RPC.

[221] Apache Software Foundation, “Apache Tomcat.” 2012.

[222] Apache Software Foundation, “Realm Configuration,” 2012. [Online]. Available: http://tomcat.apache.org/tomcat-6.0-doc/realm-howto.html. [Accessed: 01-Jan-2012].

[223] K. J. M. Janssen, a R. T. Donders, F. E. Harrell, Y. Vergouwe, Q. Chen, D. E. Grobbee, and K. G. M. Moons, “Missing covariate data in medical research: to impute is better than to ignore.,” *J. Clin. Epidemiol.*, vol. 63, no. 7, pp. 721–7, Jul. 2010.

[224] J. L. Schafer, *Analysis of Incomplete Multivariate Data*, vol. 11, no. 3. Chapman & Hall, 1997.

[225] R. Jankowski, “Implementing national guidelines at local level,” *BMJ*, vol. 322, no. 7297, pp. 1258–1259, 2001.

[226] I. Graham, M. Harrison, and M. Lorimer, Karen Piercianowski, Tadeusz Friedberg, Elaine Buchanan, “Adapting National and International Leg Ulcer Practice Guidelines for Local Use: The Ontario Leg Ulcer Community Care Protocol,” *Adv. Skin Wound Care*, vol. 18, no. 6, pp. 307–318, 2005.

[227] C. Silagy, D. P. Weller, H. Lapsley, P. Middleton, T. Shelby-James, and B. Fazekas, “The effectiveness of local adaptation of nationally produced clinical practice guidelines.,” *Fam. Pract.*, vol. 19, no. 3, pp. 223–30, Jun. 2002.

[228] M. Harrison, F. Légaré, I. D. Graham, and B. Fervers, “Adapting clinical practice guidelines to local context and assessing barriers to their use,” *Can. Med. Assoc. J.*, vol. 182, pp. E79–E84, 2010.

[229] J. A. Forsberg, J. Eberhardt, P. J. Boland, R. Wedin, and J. H. Healey, “Estimating survival in patients with operable skeletal metastases: an application of a Bayesian belief network.,” *PLoS One*, vol. 6, no. 5, p. e19956, Jan. 2011.

[230] R. E. Neapolitan, *Probabilistic Reasoning in Expert Systems*. New York: Wiley and Sons, 1990.

[231] J. Pearl, *Causality: Models, Reasoning and Inference*. Cambridge Unviersity Press, 2000.

[232] J. Cruz and D. S. Wishart, “Applications of machine learning in cancer prediction and prognosis,” *Cancer Inform.*, vol. 2, pp. 59–77, Jan. 2006.

[233] O. Gevaert, F. De Smet, D. Timmerman, Y. Moreau, and B. De Moor, “Predicting the prognosis of breast cancer by integrating clinical and microarray data with Bayesian networks.,” *Bioinformatics*, vol. 22, no. 14, pp. e184–90, Jul. 2006.

[234] N. Cruz-Ramírez, H. G. Acosta-Mesa, H. Carrillo-Calvet, L. A. Nava-Fernández, and R. E. Barrientos-Martínez, “Diagnosis of breast cancer using Bayesian networks: a case study.,” *Comput. Biol. Med.*, vol. 37, no. 11, pp. 1553–64, Nov. 2007.

[235] S. M. Maskery, H. Hu, J. Hooke, C. D. Shriver, and M. N. Liebman, “A Bayesian derived network of breast pathology co-occurrence.,” *J. Biomed. Inform.*, vol. 41, no. 2, pp. 242–50, Apr. 2008.

[236] J. Gadewadikar, E. Sarigul, O. Kuljaca, Y. Zheng, K. Agyepong, and P. Zhang, “Exploring Bayesian networks for automated breast cancer detection,” in *IEEE Southeastcon*, 2009, pp. 153–157.

[237] A. Stojadinovic, G. E. Peoples, S. K. Libutti, L. R. Henry, J. Eberhardt, R. S. Howard, D. Gur, E. a Elster, and A. Nissan, "Development of a clinical decision model for thyroid nodules.," *BMC Surg.*, vol. 9, p. 12, Jan. 2009.

[238] D. Zhao and C. Weng, "Combining PubMed knowledge and EHR data to develop a weighted bayesian network for pancreatic cancer prediction.," *J. Biomed. Inform.*, vol. 44, no. 5, pp. 859–68, Oct. 2011.

[239] J. H. Oh, J. Craft, R. Al Lozi, M. Vaidya, Y. Meng, J. O. Deasy, J. D. Bradley, and I. El Naqa, "A Bayesian network approach for modeling local failure in lung cancer.," *Phys. Med. Biol.*, vol. 56, no. 6, pp. 1635–51, Mar. 2011.

[240] A. Stojadinovic, A. Bilchik, D. Smith, J. S. Eberhardt, E. Ben Ward, A. Nissan, E. K. Johnson, M. Protic, G. E. Peoples, I. Avital, and S. R. Steele, "Clinical decision support and individualized prediction of survival in colon cancer: bayesian belief network model.," *Ann. Surg. Oncol.*, vol. 20, no. 1, pp. 161–74, Jan. 2013.

[241] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, "Data Preprocessing for Supervised Leaning," *Int. J. Comput. Sci.*, vol. 1, no. 1, pp. 111–117, 2006.

[242] N. Friedman and D. Geiger, "Bayesian Network Classifiers *," *Machine*, vol. 163, pp. 131–163, 1997.

[243] J. W. Graham, "Missing data analysis: making it work in the real world.," *Annu. Rev. Psychol.*, vol. 60, pp. 549–76, Jan. 2009.

[244] D. Rubin, "Inference and Missing Data," *Biometrika*, vol. 63, pp. 581–592, 1976.

[245] A. Dempster, N. Laird, and D. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. R. Stat. Soc.*, vol. 39, no. 1, pp. 1–38, 1977.

[246] R. F. Potthoff, G. E. Tudor, K. S. Pieper, and V. Hasselblad, "Can one assess whether missing data are missing at random in medical studies?," *Stat. Methods Med. Res.*, vol. 15, no. 3, pp. 213–234, Jun. 2006.

[247] J.-H. Lin and P. J. Haug, "Exploiting missing clinical data in Bayesian network modeling for predicting medical problems.," *J. Biomed. Inform.*, vol. 41, no. 1, pp. 1–14, Mar. 2008.

[248] G. J.W., A. Olchowski, and D. Gilreath, "How many imputations are really needed? Some ractical clairications of multiple imputation theory.," *Prev. Sci.*, vol. 8, pp. 206–217, 2007.

[249] I. Rish, "An empirical study of the naive Bayes classifier," in *IJCAI 2001*, 2001.

[250] D. W. Hosmer and S. Lemeshow, *Applied Logistic Regression*, 2nd ed. John Wiley & Sons, 2005.

[251] K. W. Kuschner, D. I. Malyarenko, W. E. Cooke, L. H. Cazares, O. J. Semmes, and E. R. Tracy, "A Bayesian network approach to feature selection in mass spectrometry data," *BMC Bioinformatics*, vol. 11, p. 177, 2010.

[252] D. Song, C. H. Ek, K. Huebner, and D. Kragic, "Multivariate discretization for Bayesian Network structure learning in robot grasping," *2011 IEEE Int. Conf. Robot. Autom.*, pp. 1944–1950, May 2011.

[253] O. Colot, P. C. Olivier, and A. El Matouat, "Information criteria and abrupt changes in probability laws," *Signal Process. Theory Appl.*, pp. 1855–1858, 1994.

[254] K. F. Rabe, S. Hurd, A. Anzueto, P. J. Barnes, S. a Buist, P. Calverley, Y. Fukuchi, C. Jenkins, R. Rodriguez-Roisin, C. van Weel, and J. Zielinski, "Global strategy for the diagnosis, management, and prevention of chronic obstructive pulmonary disease: GOLD executive summary.," *Am. J. Respir. Crit. Care Med.*, vol. 176, no. 6, pp. 532–55, Sep. 2007.

[255] E. I. George, "The Variable Selection Problem," *J. Am. Stat. Assoc.*, vol. 95, no. May, pp. 1304–1308, 2000.

[256] M. Verduijn, N. Peek, P. M. J. Rosseel, E. de Jonge, and B. a J. M. de Mol, "Prognostic Bayesian networks I: rationale, learning procedure, and clinical use.," *J. Biomed. Inform.*, vol. 40, no. 6, pp. 609–18, Dec. 2007.

[257] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "WEKA." 2009.

[258] D. D. Lewis, "Naive (Bayes) at forty: The independence assumption in information retrieval," vol. 1398, pp. 4–15, 1998.

[259] J. Zhang and K. F. Yu, "What's the relative risk? A method of correcting the odds ratio in cohort studies of common outcomes.," *JAMA*, vol. 280, no. 19, pp. 1690–1, Nov. 1998.

[260] S. Le Cessie and J. C. Van Howelingen, "Ridge Estimators in Logistic Regression," *Appl. Stat.*, vol. 41, pp. 191–201, 1992.

[261] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA: Morgan Kaufman, 1993.

[262] J. R. Quinlan, "Improved Use of Continuous Attributes in C4 . 5," *J. Artifcial Intell. Res.*, vol. 4, no. 1996, pp. 77–90, 2006.

[263] C. M. Bishop, *Pattern Recognition and Machine Learning*, vol. 4, no. 4. Springer, 2006, p. 738.

[264] A. Ng and M. Jordan, "On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes," in *Advances in NIPS 14*, 2002.

[265] J. B. . Kruskal, "On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem," vol. 7, no. 1, pp. 48–50, 1956.

[266] R. R. Bouckaert, "Bayesian Network Classifiers in Weka for Version 3-5-7," 2008.

[267] C. Wallace and K. Korb, *Learning Linear Causal Models by MML Sampling*. Heidelberg, 1999, pp. 89–111.

[268] T. Verma and J. Pearl, "An Algorithm for Deciding if a Set of Observed Independencies Has a Causal Explanation," in *Uncertainty in Artificial Intelligence*, 1992, pp. 323–330.

[269] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning Bayesian Networks: the combination of knowledge and statistical data," *Mach. Learn.*, vol. 20, no. 3, pp. 197–243, Sep. 1995.

[270] R. E. Kass and A. E. Raftery, "Bayes Factors," *J. Am. Stat. Assoc.*, vol. 90, no. 430, pp. 773–795, 199AD.

[271] D. Spiegelhalter, A. Dawid, S. L. Lauritzen, and R. Cowell, "Bayesian analysis in expert systems," *Stat. Sci.*, vol. 8, pp. 219–282, 1993.

[272] W. Buntine, "Theory refinement in Bayesian Networks," in *Uncertainty in Artificial Intelligence*, 1991, pp. 52–60.

[273] S. Yang and K. Chang, "Comparison of score metrics for Bayesian network learning," *IEEE Trans. Syst. Humans*, vol. 32, no. 3, pp. 419–428, 2002.

[274] C. K. Chow and C. N. Liu, "Approximating Discrete Probability Distributions with Dependence Trees," *IEEE Trans. Inf. Theory*, vol. 14, no. 3, pp. 462–467, 1968.

[275] D. Madigan and J. York, "Bayesian Graphical Models for Discrete Data," *Int. Stat. Rev.*, vol. 63, no. 2, pp. 215–232, 1995.

[276] M. Janzura and J. Nielsen, "A Simulated Annealing-Based Method for Learning Bayesian Networks from Statistical Data," *Int. J. Intell. Syst.*, vol. 21, no. 3, pp. 335–348, 2006.

[277] D. M. Chickering, D. E. Heckerman, and C. Meek, "A Bayesian Approach to Learning Bayesian Networks with Local Structure," in *Conference on Uncertainty in Artificial iItelligence*, 1997, pp. 80–89.

[278] R. Solomonoff, “A Formal Theory of Inductive Inference, Part I,” *Inf. adn Control*, vol. 7, no. 1, pp. 1–22, 1964.

[279] J. Ryoo, “Proof of MLE for Multinomial Distribution.” 2008.

[280] F. V. Jensen, S. L. Lauritzen, and K. Olesen, “Bayesian updating in causal probabilistic networks by local computations,” *Comput. Stat. Q.*, vol. 4, pp. 269–282, 1990.

[281] P. Mazzone, “Preoperative evaluation of the lung resection candidate.,” *Cleve. Clin. J. Med.*, vol. 79 Electro, pp. eS17–22, May 2012.

[282] A. Motohiro, H. Ueda, H. Komatsu, N. Yanai, and T. Mori, “Prognosis of non-surgically treated, clinical stage I lung cancer patients in Japan.,” *Lung cancer*, vol. 36, no. 1, pp. 65–9, Apr. 2002.

[283] T. Sobue, T. Suzuki, M. Matsuda, T. Kuroishi, S. Ikeda, and T. Naruke, “Survival for clinical stage I lung cancer not surgically treated. Comparison between screen-detected and symptom-detected cases. The Japanese Lung Cancer Screening Research Group.,” *Cancer*, vol. 3, no. 69, pp. 685–692, 1992.

[284] S. P. Riaz, M. Lüchtenborg, R. H. Jack, V. H. Coupland, K. M. Linklater, M. D. Peake, and H. Møller, “Variation in surgical resection for lung cancer in relation to survival: population-based study in England 2004-2006.,” *Eur. J. Cancer*, vol. 48, no. 1, pp. 54–60, Jan. 2012.

[285] S. Iyer, A. Roy, and A. Marchbank, “Management of Stage 1 and II Non- small-cell lung cancer at Plymouth Hospitals NHS Trust,” *Lung Cancer*, no. BTOG Abstracts, p. S70, 2013.

[286] S. Taylor, “Surgical Resection in Low stage non-small cell lung cnacer within the North Tees and Hartlepool NHS Foundation Trust,” *Lung Cancer*, vol. BTOG Abstr, p. S70, 2013.

[287] A. L. Rich, L. J. Tata, C. M. Free, R. a Stanley, M. D. Peake, D. R. Baldwin, and R. B. Hubbard, “Inequalities in outcomes for non-small cell lung cancer: the influence of clinical characteristics and features of the local lung cancer service.,” *Thorax*, vol. 66, no. 12, pp. 1078–84, Dec. 2011.

[288] K. Lau, S. Rathinam, D. Waller, and M. D. Peake, “The Effects of Increased Provision of Thoracic Surgical Specialists on the Variation in Lung Cancer Resection Rate in England,” *J. Thorac. Oncol.*, vol. 8, no. 1, pp. 68–72, 2013.

[289] F. G. Cozman, “JavaBayes.” University of Sao Paulo, Sao Paulo, 2001.

[290] F. G. Cozman, “The Interchange Format for Bayesian Networks,” *Carnegie Mellon University*, 2001. [Online]. Available: http://www.cs.cmu.edu/~fgcozman/Research/InterchangeFormat/. [Accessed: 01-Oct-2011].

[291] N. L. Zhang and D. Poole, “Exploiting causal independence in Bayesian network inference,” *J. Artif. Intell. Res.*, pp. 301–328, 1996.

[292] F. G. Cozman, “EBayes.” Sao Paulo, 1999.

[293] J. Nyberg, J. Brian, G. Marcot, and R. Sulyma, “Using Bayesian belief networks in adaptive management,” *Can. J. For. Res.*, vol. 3116, pp. 3104–3116, 2006.

[294] D. K. Owens, R. D. Shachter, and R. F. Nease, “Representation and analysis of medical decision problems with influence diagrams.,” *Med. Decis. Making*, vol. 17, no. 3, pp. 241–62, 1997.

[295] R. D. Shachter, “Probabilistic Inference and Influence Diagrams,” *Oper. Res.*, vol. 36, no. July-August, pp. 589–605, 1988.

[296] A. Rector, “AIM: a personal view of where I have been and where we might be going.,” *Artif. Intell. Med.*, vol. 23, no. 1, pp. 111–27, Aug. 2001.