Title

Using knowledge encoded in graphical disease models to support context -sensitive visualization of medical data

Author

Hsu, W

Publication Date

2009

Peer reviewed

UNIVERSITY OF CALIFORNIA

Los Angeles

Using Knowledge Encoded in Graphical Disease Models to Support Context-Sensitive Visualization of Medical Data

A dissertation submitted in partial satisfaction of the

requirements for the degree Doctor of Philosophy

in Biomedical Engineering

by

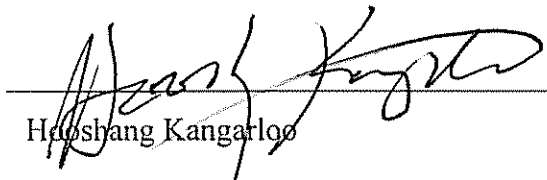
William Hsu

2009

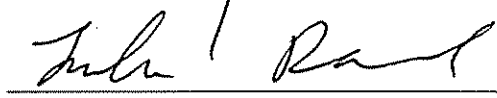
For my family and friends—oh, the places we'll go!

The dissertation of William Hsu is approved.

Denise Aberle



Hooshang Kangarloo



Judea Pearl

Alex A.T. Bui, Committee Chair

University of California, Los Angeles

2009

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to express my utmost gratitude and respect to my advisor, Professor Alex Bui. My intellectual growth and passion for biomedical informatics is a result of his guidance. He taught me to be meticulous in everything that I do and how to transform plain Microsoft Visio and PowerPoint diagrams into works of art. I would also like to thank Professor Denise Aberle for her insights and dedication to helping students like me appreciate the beauty of medicine. I have great admiration for Professor Judea Pearl; his enthusiasm is contagious, and his ability to turn evil into good has taught me more about life than I could ever learn in academia. And last but certainly not least, I am in debt to the kindness and mentorship of Professor Hooshang Kangarloo, who taught me how to polish my technical nugget, sharpen my elevator story, and never settle for average, because "average is the worst enemy to excellence." I am eternally thankful and honored to have been their student.

During the five years of my graduate life, I have had the privilege of working with some of the brightest and most talented people that I know. I am grateful to the entire faculty at UCLA Medical Imaging Informatics for sharing their expertise. I would like to acknowledge Professor Ricky Taira, who is a never-ending source of ideas and inspiration. He also never fails to make me laugh. I also would like to thank Professors Craig Morioka, Jim Sayre, Suzie El-Saden, and Usha Sinha, who have been instrumental towards helping me mature as a researcher. I would like to thank the members of the staff who have gone above and beyond what they needed to do: Isabel Rippy,

Kerry Engber, Susan Barretta, Greg Weinger, Brandon Bigby, and Gregg Hamm. And above all, I am so thankful for all of my peers—past and present—who have laid the foundation for my work, fostered my competitive spirit in fantasy basketball, generously shared their snacks and SPKs, beaten me at first person shooters, and most of all, graced me with their friendship: Dr. Tin Man Lee, Dr. Luis Selva, Emily Watt, Corey Arnold, Lew Andrada, Dr. Vijay Bashyam, Eugenio Iglesias, Neda Jahanshad, Dr. Hussain Tameem, Dr. Siamak Ardekani, Dr. Len D'Avolio, Kyle Singleton, Brian Burns, Jean Garcia, Dr. Winston Wu, Galya Rees, Michael Tseng, Dr. Shishir Dube, Dr. Frank Meng, Jen Garcia, Dr. Alicia Dordoni, and Dr. Roderick Son. I would also like to thank my dear classmates in Biomedical Engineering IDP who I have had the pleasure of knowing for all these years.

I want to express my gratitude to the brothers and sisters at the Garden Christian Fellowship for their friendship and prayers throughout my studies. I have been blessed with everyone in my community groups; they constantly remind me that there is more to life than writing. I am particularly thankful to Lois Wee, Rosa Chen, Perry Li, and Sunni Han.

I want to specially thank Sandy Chan for her humor, love, and patience.

I am forever grateful for the support of my family and especially my parents, Tsung-Yuan and Ingrid Hsu. They have always encouraged me to dream big and sacrificed so much to give me the opportunity to pursue those dreams.

And finally, I want to thank God because He makes all things possible. Psalm 115:1: "Not to us, O Lord, not to us, but to your name be the glory, because of your love and faithfulness."

Portions of Chapter 2 are based on content from "Chapter 4: Medical data visualization: Toward integrated clinical workstations" by Bui, AAT and Hsu, W in Medical Imaging Informatics by Bui, AAT and Taira RK, eds.

This work has been supported in part by the NIH NLM Training Grant T15-LM007356 and NIBIB R01-EB000362.

VITA

November 26, 1982	Born, Darien, Illinois
2004	B.S., Biomedical Engineering Johns Hopkins University Baltimore, MD
2004-2008	National Library of Medicine Predoctoral Fellow Biomedical Engineering IDP Medical Imaging Informatics Group University of California, Los Angeles
2008-2009	Graduate Student Researcher Biomedical Engineering IDP Medical Imaging Informatics Group University of California, Los Angeles
2008-2010	Student Editorial Board Member Journal of the American Medical Informatics Association

PUBLICATIONS AND PRESENTATIONS

Bashyam V., **Hsu W.**, Watt E.W., Bui A.A.T., and Taira R.K. A Problem-centric Organization and Visualization of Patient Imaging and Clinical Data. RadioGraphics. 2009 March; 29(2): 331-43. *(Invited paper)*

Hsu W., Aberle D.R., El-Saden S.M., Kangarloo H., Bui A.A.T. A Visual Query Application for Interacting with a Probabilistic Model of Tumor Image Features. Presented at RSNA Annual Meeting, Chicago, IL. December, 2007.

Hsu W., Antani S.K., Long L.R., Neve L., and Thoma G.R. SPIRS: A Web-based Image Retrieval System for Large Biomedical Databases. Int J Med Inform. 2009 April; 78(Suppl 1): S13-24. *(Invited paper)*

Hsu W. and Bui A.A.T. A Framework for Visually Querying a Probabilistic Model of Tumor Images. In: Proc. AMIA Symp; 2006. p.354-358. *(Finalist, Student Paper Award)*

Hsu W., El-Saden S.M., and Bui A.A.T. A Visual Query Interface for Assisting in Decision Support of Tumors Using Image Findings Structured by Bayesian Networks. Am J Roentgenol. 2006 April;186(4 Suppl):A30-2.

Hsu W., Long L.R., and Antani S.K. SPIRS: A Framework for Content-based Image Retrieval from Large Biomedical Databases. In: Proc. 12th World Congress on Health (Medical) Informatics; 2007. p.188-92. *(Nominated, Best Paper Award)*

Hsu W., Taira R.K., Bui A.A.T., Kangarloo H. Chapter II: Integrating Imaging and Clinical Data for Decision Support. In: Exarchos T.P., Papadopoulos A., Fotiadis, D.I.: Handbook of Research on Advanced Techniques in Diagnostic Imaging and Biomedical Applications. Hershey, PA: IGI Global; 2009. p.18-33.

Hsu W., Watt E.W., Kangarloo H., Bui A.A.T. Tailoring the Presentation of an Imaging-based Electronic Medical Record Based on Content and Task. Presented at RSNA Annual Meeting, Chicago, IL. December, 2008.

Taira R.K., Bui A.A.T., **Hsu W.**, Bashyam V., Watt E.W., Andrada L., El-Saden S., Cloughesy T., Kangarloo H. A Tool for Improving the Longitudinal Imaging Characterization for Neuro-Oncology Cases. In: Proc. AMIA Symp; 2008. p.712-6.

Taira R.K., **Hsu W.**, Bashyam V. A Case Retrieval System for Neuroradiology Using a Natural Language Processing Engine and Treemap Visualization. Presented at RSNA Annual Meeting, Chicago, IL. December, 2008. *(Awarded,* C*ertificate of Merit)*

ABSTRACT OF THE DISSERTATION

Using Knowledge Encoded in Graphical Disease Models to Support Context-Sensitive Visualization of Medical Data

by

William Hsu

Doctor of Philosophy in Biomedical Engineering

University of California, Los Angeles, 2009

Professor Alex A.T. Bui, Chair

Given the large quantity of diverse, heterogeneous data in a typical patient record, users spend much of their time and effort finding relevant information to help accomplish their tasks. One of the greatest problems in today's healthcare environment is matching the increased capability of gathering patient data with a comparable ability to understand, analyze, and act rationally upon this information. This dissertation attempts to bridge this gap by presenting methods for creating context-sensitive visuali-

zations using graphical disease models. Building upon past efforts in Bayesian belief networks and explanation generation, this work explores how the model's variables, defined relationships, and probabilities are used to identify which data elements in the patient record are important and how the information should be presented for a particular context. A data model, called the visual dictionary, is used to integrate contextual information from graphical disease models and other knowledge sources (*e.g.*, ontologies and user/task models) to generate instructions for laying out patient data in a graphical user interface. These concepts are implemented in two separate applications that demonstrate how context-sensitive visualizations can: 1) be applied towards helping users query large biomedical repositories; and 2) generate an integrated, longitudinal view of a multimedia patient record. The applications were used to evaluate the feasibility of using graphical disease models to retrieve relevant documents and to obtain feedback on the adaptive interfaces through pilot usability studies. Initial results from the pilot studies were positive overall. Developing context-sensitive visualizations that facilitate users with querying these models and understanding the results is a significant step towards using collected data to improve patient care at the bedside.

CHAPTER 1

Introduction

1. Overview

Graphical models are becoming increasingly popular for modeling complex phenomena and reasoning under uncertainty in many domains, including medicine. In addition to being able to answer a wide range of probabilistic queries, graphical models can be used to explain how a certain conclusion is obtained, providing insight into how variables in these models interact with each other. This dissertation expands upon explanation generation with the purpose of identifying a subset of data that is relevant for a given situation (*e.g.*, consultation with the primary care physician) and adapting the presentation of that data based on the changing context. Visualization is an important aspect of the medical decision making process because providing relevant patient information in a timely manner is critical to improving a physician's ability to diagnose and treat a patient. This work exploits properties of the graphical model, such as its variables, encoded relationships, and parameters, to determine how data in a patient's record can be integrated and presented in a clinical display. This process involves three steps: 1) determining what data elements from the patient record are relevant to a user for a given situation; 2) influencing how data elements are mapped to visual representations; and 3) dictating how multiple visual representations are combined and

laid out on the screen. The result is a context-sensitive display that tailors how information is presented based on medical condition, user, and task. This introductory chapter provides the motivation for this dissertation research, outlines the aims of this work, and summarizes its contributions.

1.1. Background and Motivation

Medicine has benefited from improvements in diagnostic technologies that provide an unprecedented amount of data spanning a wide spectrum of types (*e.g.*, clinical documents, medical images) and scales (*e.g.*, molecular, cellular, tissue). For instance, gene microarray experiments enable the analysis of thousands of genes simultaneously, providing a detailed picture of a patient's susceptibility to certain diseases. However, it remains a challenge to integrate and understand this data in the context of the entire patient record. Ongoing efforts to create an electronic medical record (EMR) that digitally stores data collected during patient encounters will only increase the amount of information available [5]. However, having large amounts of data that are electronically accessible do not necessarily translate to improved physician efficiency, reduced costs, or ultimately, better patient outcomes. A challenge in today's healthcare environment is matching the increased capability of gathering patient data with a comparable ability to translate this knowledge into improved care at the bedside. One hurdle is that much of the data does not lend itself to formal, automated analysis: additional steps are needed to abstract pertinent knowledge from the data. Along this front, new techniques for abstracting this data using statistical and probabilistic models have been

developed, allowing users to perform classification and prediction tasks based on a given individual's data.

In this work, I focus on a class of probabilistic models called *graphical models*, and a particular subset of graphical models called *Bayesian belief networks* (BBN). One drawback of using BBNs—and graphical models, in general—in clinical practice is the lack of an integrated and intuitive interface that assists users with querying the model and understanding the results. Currently, these models are commonly visualized as a graph where highlighted nodes correspond to selected variables. While a physician could be trained to manipulate and understand this representation, this interface is a departure from the traditional graphical user interface (GUI) of an EMR. Rather than visualize the graph directly, I use the BBN to influence how the EMR is displayed on the screen. My work leverages properties of the graphical disease model to identify influential nodes and paths in the model for a given context and then use a data model called the *visual dictionary* to translate this explanation into a visual display.

1.2. Overview of Context-sensitive Visualization

Given the wealth of patient data, visualization of the medical record is critical to helping physicians understand and act on the available information. While different paradigms (*e.g.*, source-oriented, problem-oriented, time-oriented; see Section 2.4.1) have been suggested for integrating and laying out patient data, the strengths of each approach need to be combined into one truly comprehensive visualization of the medical record. One aspect of the user interface that has yet to be explored is the role that con-

text plays in influencing how the data is displayed. *Context* is defined as information that provides a better understanding of an entity and the situation that it is in [6]. Here, I define an entity as a data element in the patient record. Context can therefore be represented as answers to questions such as *why was the data element recorded; how was the data element recorded; what other entities is the element related to; and who should be able to access it?* The utility of a user interface is dependent in its ability to provide the user with the tools not only for finding and viewing relevant parts of the patient record quickly but also for understanding the context that supplements the user's understanding of the situation. Taking a context-sensitive approach may help users identify cause-effect relationships that would not be apparent in information visualizations alone. Bui et al. [7] first explored the concept of a context-sensitive visualization in medicine; their interface, TimeLine, grouped multiple, simpler visualizations together and generated a new composite presentation based on how the user interacted with the system.

The process of generating a context-sensitive visualization from patient data can be decomposed into several steps: *characterizing* the data that is available in the patient record and able to be displayed; *identifying* the data that needs to go into the display; *prioritizing* the data based on relevance to the user's actions; *relating* the data elements; *selecting* the appropriate visual metaphor for the data; and finally, *laying out* the visual metaphors. This process is summarized in Figure 1.1. While context may be obtained from various sources (*e.g.*, user interaction, task model, knowledge sources),

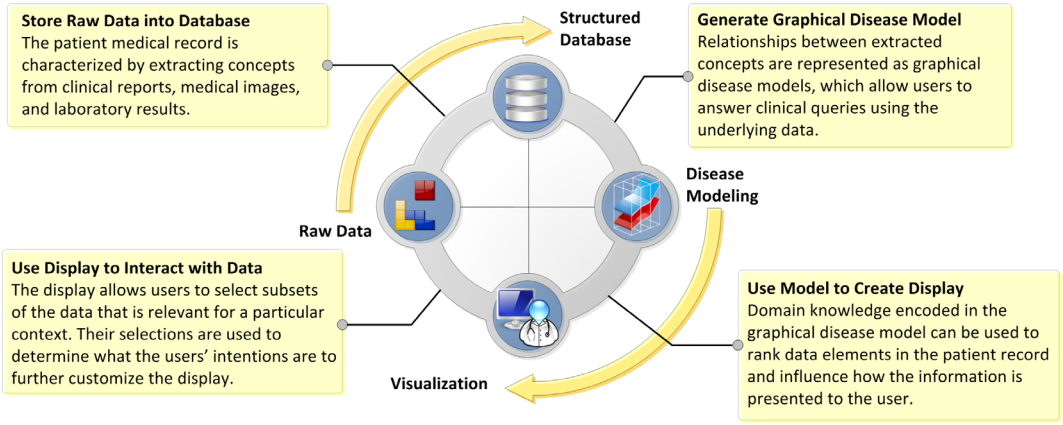


Figure 1.1: A framework for context-sensitive visualization. Patient data is first characterized using information extraction techniques and stored in a relational database. The structured data is used to generate graphical disease models, which provide information for tailoring the user interface based on changing contexts. As the user interacts with the interface, the context is re-evaluated, and the display is updated.

this work primarily explores obtaining context from graphical disease models by extending algorithms to analyze aspects of the model to generate rules that guide how information is presented on the screen.

1.3. Aims of the Dissertation

A patient record typically contains a large amount of diverse data collected over the course of a chronic illness or even a patient's lifetime. While visualizations provide tools for viewing and interacting with this data, the task of filtering irrelevant information and linking disparate pieces of evidence together remains the user's responsibility. Much time and effort is spent searching through the data to find evidence that is relevant to diagnosing or treating a patient. My work attempts to address this problem by using context-sensitive visualization as a way to filter and adapt the presentation of patient data. This goal is accomplished through two subtasks:

1. Using knowledge encoded in the structure and attributes of graphical disease models to determine the relevance of a given piece of information in the patient record; and

2. Integrating knowledge from the model with characterizations of the data and available visualizations to influence how information is rendered on the display.

The first subtask seeks to leverage domain knowledge in graphical disease models as a way to filter and relate data elements in the patient record. My dissertation explores several algorithms for traversing a graphical model and characterizing the relationships that exist among variables. I also examine how other knowledge sources such as ontologies and user/task models may be used to further supplement the information provided by a disease model.

The second subtask results in the development of a visual dictionary. The visual dictionary is a data model that relates data elements with composition rules and graphical attributes (*e.g.*, transparency, layering) to influence how data elements are prioritized and rendered on the screen. My work uses the visual dictionary to integrate knowledge from disease models and other knowledge sources and to match data elements with appropriate visualizations based on their characterizations.

Two applications have been implemented to demonstrate the utility of context-sensitive visualization in clinical practice: a visual query interface (VQI) [8], which

provides clinicians with the ability to query large repositories of biomedical imaging data; and an adaptive electronic health record (AdaptEHR) viewer, which selectively presents patient data in a longitudinal display based on context. In VQI, the user can search a large database of images and clinical data by posing queries using graphical metaphors that visually represent variables and states of the underlying disease model. The structure of the model is used to guide the user through the query formulation process. In AdaptEHR, the visual dictionary is leveraged to generate a summary display of patient data for a given medical problem, user, and task. The disease model is used to guide the identification of specific data elements (*e.g.*, documents, test results) that are the most relevant to the selected disease, and the composition rules are used to combine and lay out related elements.

1.4. Summary of Contributions

The major contributions of this work are:

- *Algorithms for characterizing properties of the graphical disease model as quantitative measures of relevance and influence.* Building upon previous work in generating explanations from BBNs [9-12], I characterize disease models by examining their variables, structures, and parameters to determine how data elements in the patient record relate to one another and ultimately, how they are to be presented to the user. First, variables in the disease model are mapped to corresponding data elements in the patient record using the process of query expansion; this process is used to *identify* the data elements that will be displayed. Then, the algo-

rithms for computing paths of influence, influential evidence [11], strength of influence [9], and value of information [12] are applied towards *filtering* and *prioritizing* elements in the patient record based on changing contexts. Section 3.2 details the approach for using graphical models to determine relevancy and influence of variables for a particular context.

- *An approach for integrating disparate knowledge sources (e.g., ontologies, user/task models) to provide additional context for customizing a display.* Ontologies such as the Unified Medical Language System (UMLS) [13], National Cancer Institute (NCI) Thesaurus [14], and RadLex [15], provide rich sources of knowledge about a concept and how it relates with other terms in a domain. My work uses the terms, semantic types, and relationships defined in these ontologies to *relate* data elements with one another. I also explore how user models help elucidate the user's information needs. Section 3.4 of this dissertation provides a discussion on how these knowledge sources are used to provide context.

- *A visual dictionary to translate contextual information into rules that influence how data is presented on the screen.* The visual dictionary integrates contextual information provided by the graphical disease model and other knowledge sources into a set of rules that dictate how the display is generated. The visual dictionary first *selects* an appropriate visual metaphor for each data element that has been identified for display. A set of composition rules is then used to *lay out* visual me-

taphors on the screen. The visual dictionary and composition rules are detailed in Section 3.5.

- *Two clinical applications that use graphical disease models and the visual dictionary to adapt the presentation of the medical record based on context.* Working in collaboration with radiologists, I have created two different applications that address practical problems of information retrieval and visualization of multimedia patient records. The first application, visual query interface, demonstrates how the BBN can adapt the user interface to guide users with formulating complex queries to the underlying disease model and how the BBN is applied towards retrieving similar cases. The second application, an adaptive electronic health record viewer, demonstrates how the BBN can be used to highlight and spatially organize patient information based on how relevant it is to the user and task. These applications are discussed in Chapter 4.

1.5. Organization of the Dissertation

The remainder of this dissertation is organized as follows. Chapter 2 provides a brief tutorial on the theoretical foundations of graphical models and reviews related work in developing knowledge sources and medical visualization. Chapter 3 describes the algorithms used to extract knowledge encoded in graphical models and discusses the visual dictionary. Chapter 4 details the implementation of two prototype applications: VQI and AdaptEHR. Results from a pilot evaluation are briefly presented in Chapter

5. Finally, Chapter 6 concludes with a discussion of the limitations of this work and future directions.

CHAPTER 2

Background

2. Overview

My work in context-sensitive visualization builds upon past work done in the areas of graphical models, knowledge representation, and information visualization; this chapter surveys developments in each area. The first section lays a theoretical foundation for understanding graphical models. Section 2.2 surveys other sources of knowledge that can be used to provide context: ontologies and user/task models. Section 2.3 reviews existing work on medical visualizations by expanding upon the object-oriented taxonomy described in [16]. Finally, Section 2.4 presents related efforts towards creating a longitudinal, integrated display for viewing the electronic medical record.

2.1. Graphical Disease Models

The creation of disease models poses several challenges [17]: 1) the uncertainties inherent to medical knowledge must be captured; 2) the models need to be sufficiently intuitive so that domain experts (*e.g.*, physicians) can understand the explanations proposed by the system; and 3) the models must be practically analyzable by algorithms to support querying. While systems using propositional logic (*e.g.*, rule-based) are capable of encoding medical knowledge (*e.g.*, if patient exhibits a seizure, then the patient may have a brain tumor), they are not able to handle exceptions that have not

been defined in the knowledge base (*e.g.*, a seizure is not always a result of a brain tumor; it could be related to epilepsy). [17] discusses three approaches that have been explored to overcome this limitation: 1) augmenting the set of rules to handle exceptions; 2) replacing the hard truth value associated with each proposition with a generalized numeric truth value that captures the confidence in the rule; or 3) approaching the problem from a perspective using probability theory. Among the available reasoning approaches, probability theory is particularly well-suited for modeling clinical observations because of its clear semantics in terms of beliefs or frequencies, and its ability to take advantage of a problem's structure to reduce the computational burden of reasoning without giving up clarity and correctness. When coupled with graph theory, the resulting representation, called a *graphical model*, provides a qualitative language for probabilistic and causal notions that can be understood and manipulated visually rather than having to perform the equivalent algebraic calculations of probability distributions. Graphical models provide an intuitive way of representing and visualizing the relationships among many variables. While probabilistic models may be formulated and solved using purely algebraic manipulation, the graph representation gives insight into the properties of the model, such as conditional independence relationships between the variables.

A graph is comprised of nodes and connected by edges. Each node represents a random variable, and each edge represents probabilistic relationships between nodes[1].

Definition: A graph G is defined by a collection *X* of nodes and a set E of edges between those vertices.

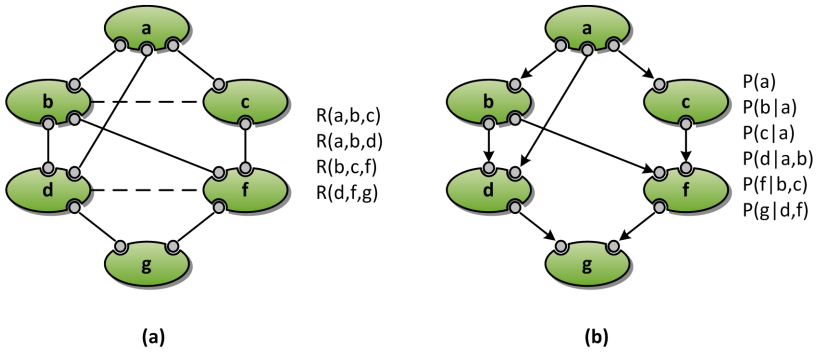


Figure 2.1: Two types of graphical models exist: **(a)** An undirected Markov random field. **(b)** A directed acyclic graph called a Bayesian belief network.

Two main types of graphical model are depicted in Figure 2.1: undirected (*e.g.*, Markov random fields) and directed (*e.g.*, Bayesian belief networks).

Definition: An edge $e = (i, j)$ connecting nodes i and j is *undirected* if it is symmetric with respect to i and j and $(i, j) = (j, i)$.

Definition: An edge $e = (i, j)$ connecting nodes i and j is *directed* if it starts at i and ends at j and $(i, j) \neq (j, i)$.

[1] To differentiate between variable names and the concepts that they represent, variable names are written in Arial.

Definition: When there is a direct link $X_i \rightarrow X_j$ from X_i to X_j then X_i is said to be the parent of X_j and X_j is the child of X_i.

Definition: A directed graph is said to be cyclic if it has at least one cycle, otherwise it is acyclic.

The user may pose queries to the graph by selecting and specifying values to its nodes. A node becomes *instantiated* when the user sets the variable to a specific value. Nodes that have been selected by the user have special semantics; they can either be classified as an evidence node or a target node.

Definition: Evidence nodes represent variables that have been instantiated by the user.

For example, for the model depicted in Figure 2.2, the node smoking can be instantiated as one of two states; true or false. When a user specifies that smoking = true, then the node is called an evidence node.

Definition: Target nodes represent variables whose outcomes are of particular interest to the user.

For instance, emphysema is a logical target node for users who desire to predict whether a patient has emphysema given whether or not he/she smokes, has a cough, or has asthma.

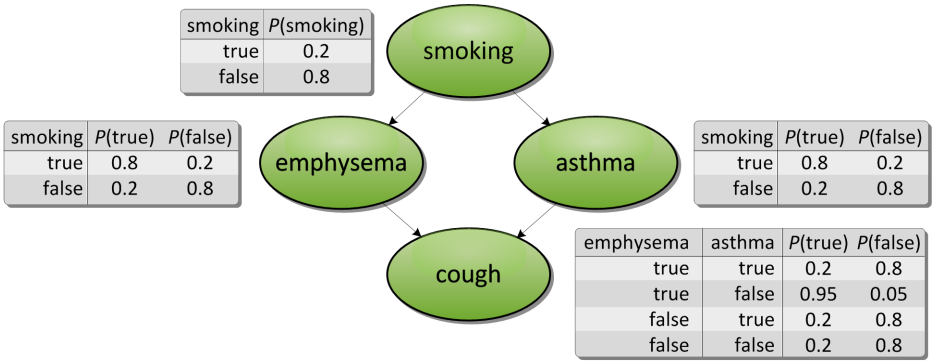


Figure 2.2: A directed graphical model called a Bayesian belief network that represents the effects of smoking.

2.1.1. Bayesian belief networks

A Bayesian network is an instance of a graphical model whose edges are directed. Mathematically, a Bayesian network is a directed acyclic graph (DAG) whose nodes represent random variables and edges that represent associations, or in some cases, causal links between nodes. The edges in the DAG represent the probabilistic influences between the variables. The absence of links in the graph conveys interesting information about the properties of the class of distributions represented by the graph. A variable X_i is dependent on its parents and children in the DAG but is conditionally independent of any of its non-descendents given its parents, a property known as the Markov condition.

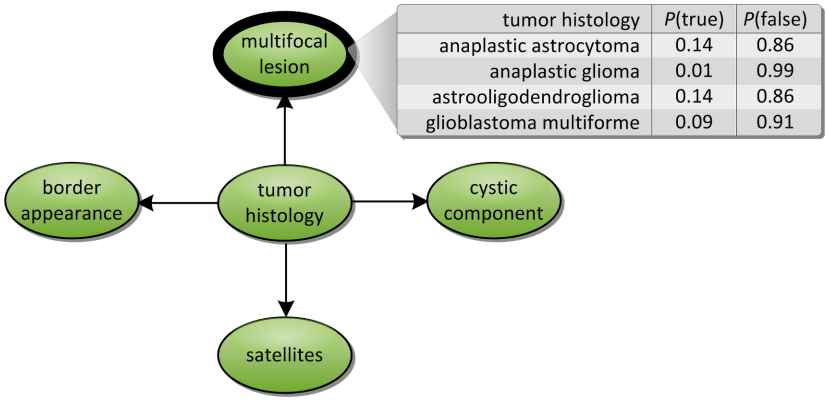


Figure 2.3: A Bayesian belief network with the conditional probability table for multifocal lesion shown.

The structure of the directed graph represents a factorization of the joint probability distribution. Consider a joint probability distribution P(U) where U is a set of variables $X_1 \ldots X_n$ that are represented in the DAG. Each variable X_i in the DAG is specified by a set of conditional probability distributions: P(X | parents(X)). Each of these distributions describes the joint effect of a specific combination of values for the parents of X_i on the probability distribution over the values of X_i. These sets of conditional probability distributions define a unique joint probability distribution that is factorized over the graph's topology using the following equation:

$$P(U) = \prod_X P(X \mid parents(X))$$

This equation is called the Markov factorization of P(U). Figure 2.3 illustrates an example Bayesian network that models how the histological grade of a brain tumor relates to other pathological features. When the conditional probability table of a single variable—in this case, multifocal lesion—is examined, only a subset of the probabilities is shown. Nevertheless, if all of the conditional probability tables in the model are

multiplied together, the full joint probability distribution is obtained. This property enables the creation of computationally manageable models: without conditional independencies, the full joint probability distribution requires $O(2^n)$ probabilities for n binary variables, but in comparison, the factorized model requires only $O(n2^k)$ where k is the maximum number of parents feeding into a node.

The primary application of disease models is to answer clinically relevant questions using the patient's data. Several different types of queries can be posed to a BBN: probability of evidence, posterior marginals, most probable explanation (MPE), and maximum a posteriori (MAP). The reader is referred to [18] for a comprehensive review of these algorithms. In my work, two types of queries, MPE and MAP, are primarily used and are described further below:

- **Most probable explanation (MPE)**: The objective of an MPE query is to identify the most probable instantiation of the entire network (*i.e.,* the state of all evidence variables) given some evidence. For example, from Figure 2.2, an MPE query may ask the following: *given that the patient is a smoker and presents with a cough, what is the most likely state of the remaining evidence variables (i.e., patient has emphysema, patient has asthma)?* A specific statement of an MPE query is as follows: let $X_1, X_2, ..., X_n$ represent the evidence variables, and $\boldsymbol{e}$ the provided evidence, then an MPE attempts to find an instantiation of $x_1, x_2, \ldots, x_n$ such that $P(x_1, x_2, \ldots, x_n \mid \boldsymbol{e})$ is maximized. Note that the most probable explanation cannot be obtained directly from individual conditional probabilities: if $x_1, x_2, \ldots, x_n$ are chosen

to maximize P(x_i | ***e***) rather than the global problem, then the choice of x_i is not necessarily the most probable explanation.

- **Maximum *a posteriori* (MAP):** Let **M** be some subset of variables in the belief network, and ***e*** some evidence; the objective of a MAP query is to find an instantiation of ***m*** such that P(***m*** | ***e***) is maximized. To illustrate, consider Figure 2.2 again; a MAP query to this model would be: *what the most likely state for* emphysema *is given that the patient presents with a cough*? Note that this query does not attempt to provide information on asthma or smoking. MPE is actually a special case of a more general type of query that attempts to find the most probable instantiation for a subset of network variables (MPE, therefore, is simply the situation when the subset is the entire set of evidence variables in the network).

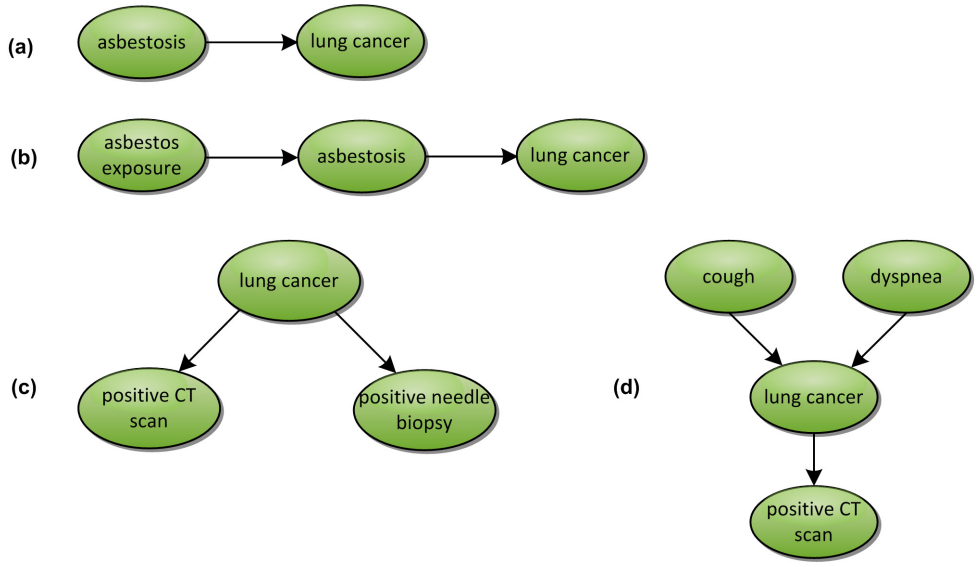


Figure 2.4: Different ways that variables can be connected in a directed acyclic graph: **(a)** direct connection; **(b)** serial connection; **(c)** divergent connection; and **(d)** convergent connection.

2.1.2. D-separation

The DAG structure of the BBN allows us to follow the flow of information from one variable to all other variables in the network. There are four ways that information is transmitted through a variable; these are illustrated in Figure 2.4. Each case is examined to determine whether any pair of variables is independent given the evidence entered into the model using a criterion called *d-separation*. The formal definition is as follows:

Definition: If X, Y, and Z are three disjoint subsets of nodes in a DAG D, then Z is said to d-separate X from Y, denoted $<X \mid Z \mid Y>_D$, if there is no path between a node in X and a node in Y along which the following two conditions hold: 1) every node with converging arrows is in Z or has a descendent in Z; and 2) every other node is outside Z [12].

The following four cases exemplify how d-separation works:

- **Direct connection.** If X and Y are connected by an edge, then X and Y are dependent, under the empty condition. As illustrated in Figure 2.4a, knowing that the patient has been exposed to asbestos will increase the belief that he is at risk of lung cancer.

- **Serial connection.** Given a series of nodes, X, Z, and Y, where X is connected to Y through Z, if Z is not observed, X and Y are dependent. In other words, evidence may be transmitted through a serial connection unless the state of the variable in

the connection is known. As in the previous example, knowing that the patient has been exposed to asbestos increases the belief that he is at risk of lung cancer. However, if it is already known that the patient has asbestosis, then knowing that he has been exposed to asbestos becomes redundant (Figure 2.4b).

- **Diverging connection.** If Z is the parent node of X and Y, and Z is not observed, X and Y are dependent. In Figure 2.4c, knowing that the patient has a positive computed tomography (CT) scan will increase the belief that he also has a positive needle biopsy. However, if it has already been confirmed that the patient has lung cancer, knowing that he has a positive CT scan will not increase the belief of a positive result from a biopsy.

- **Converging connection.** In the case where Z is the child of X and Y, neither Z nor any of its descendants are observed, X and Y are independent. For instance in Figure 2.4d, if the patient is known to have lung cancer, knowing the patient has a cough will increase the belief that he also has dyspnea through a process called explaining away. Also, without knowing that the patient has lung cancer, if the patient has a positive CT scan, then this knowledge increases the belief that he has lung cancer and therefore makes the beliefs of cough and dyspnea dependent.

The concepts of conditional independence and d-separation are important for determining whether an arbitrary node in the model is relevant to a specified target node. Based on this property, the disease model can be used as a source of determining

whether a data element is relevant for a particular context; Section 3.2 explores this problem in further detail.

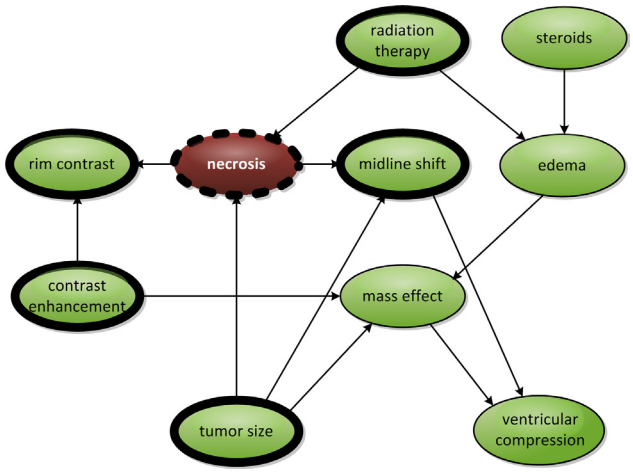


Figure 2.5: The Markov blanket of the node necrosis (dotted outline) consists of the node's parents, children, and spouses (solid outlines). Given knowledge about the nodes in the Markov blanket, all other nodes in the network are conditionally independent of necrosis.

2.1.3. Markov blanket

While the previous section describes a generalized rule for identifying conditional independencies in the model using d-separation, this section deals with a special case called the *Markov blanket*. A Markov blanket is a set of nodes comprised of the target node's parents, children, and spouses. When instantiated, the Markov blanket isolates the target node from the rest of the network. Consider the example of a Markov blanket depicted in Figure 2.5. If the selected node is necrosis, its Markov blanket, according to the model, is comprised of: radiation therapy, tumor size (parents); rim contrast, midline shift (children); and contrast enhancement (spouses). When evidence is pro-

vided for all of the variables in necrosis' Markov blanket, then necrosis is independent of steroids, edema, mass effect, and ventricular compression.

2.1.4. Extensions of BBNs

Extensions to the traditional Bayesian belief network framework have been explored to provide the ability to encode additional semantics. Two of them are reviewed here: annotated BBNs and influence diagrams.

2.1.4.1. Annotated BBNs

Additional metadata about the model may be stored about the model either as part of the file structure in which the model is saved or using external sources. Metadata can include information about assumptions made when the model was constructed, constraints for using the model, or links to published literature that support the structure and parameters specified in the model. For example, an annotated Bayesian belief network [19] provides a framework for appending additional contextual information to each variable. Annotations are useful for adding semantic knowledge such as encoding the concept unique identifier that links the variable to a standard concept in a medical lexicon. Additional information about the node color, position, and type can also be retained. Node color is a potential method for specifying user-defined groups of variables, which can be used by an application to recall multiple related variables. This information is then used to determine how a variable is handled or classified. In my work, annotations are used to encode information that maps continuous values ex-

tracted from the patient record to discrete states associated with each variable; annotations are discussed further in Section 3.2.1.3.

2.1.4.2. Influence diagram

Influence diagrams are a generalization of Bayesian networks: not only do they encode knowledge about a domain, but they are also able to select a strategy that maximizes the chance of a desired outcome by weighing each choice with a cost. Influence diagrams permit different configurations of this model and potential choices to be considered in terms of quantifiable values supplied via a *utility function, U*(*a*), where *a* represents an action. The aim, therefore, is to identify the configuration and actions that maximize the utility functions that solve $\text{argmax}_{\mathbf{A}} \sum U(\mathbf{x},\mathbf{a}) P(\mathbf{x} \mid \mathbf{e})$ where A represents a set of actions. Influence diagrams consist of nodes and edges as their graphical model counterparts but reclassify the nodes into three types:

- Chance nodes, which appear as ovals, are random variables, similar to the evidence variables in a BBN. Like evidence variable nodes, CPTs are associated with chance nodes. Chance nodes can have both decision and other chance nodes as parents.

- Decision nodes, which appear as rectangles, represent those points in the state/process where a choice of actions can be made; the result of a decision is to influence the state of some other variable (*e.g.,* a chance node). An influence diagram must contain at least one decision node.

- Utility nodes, which are represented as diamonds, measure the overall "outcome" state, with the goal of optimizing the utility (*i.e.,* maximizing) based on the contributing chance, decision, and causal factors.

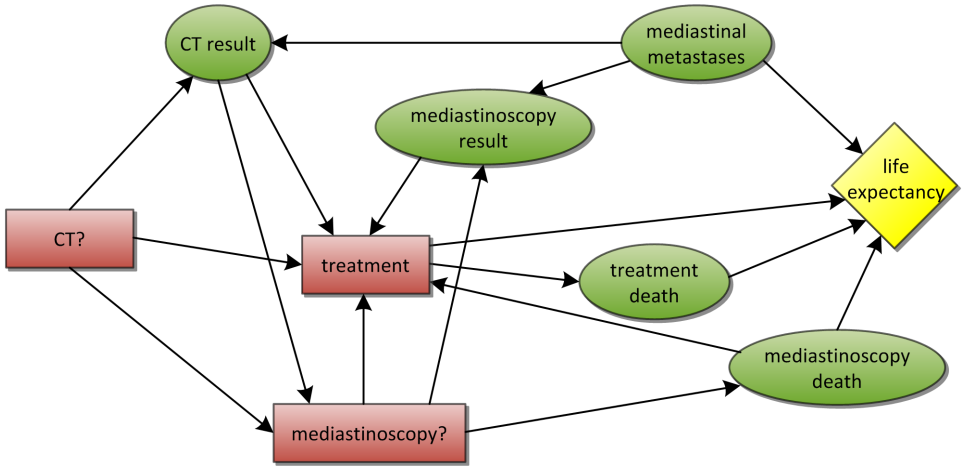


Figure 2.6: An influence diagram for determining whether a patient has a lung metastasis and what the optimal choice of treatment would be. The example is reproduced from [4].

Additionally, some influence diagrams include *deterministic nodes* (drawn as a double oval), which are defined as nodes with constant values or algebraically calculated from parent nodes' states. Once the parent nodes are known, the child node's state is definitively assigned. The paths define the sequence in which decisions are made. An example influence diagram is shown in Figure 2.6.

Influence diagrams are a key representation used in evidence-based medicine (EBM). An underlying principle of EBM is that decisions take into consideration an individual's preferences (*e.g.,* with respect to diagnostic and treatment options): by fixing the selection within a decision node, an influence diagram can view a patient's prefe-

rences as an explicit constraint within the optimization problem. The utility node can be seen as being related to a patient's quality of life (*e.g.,* for decisions involving substantial risk, quality-adjusted life expectancy, QALE) in addition to considering cost and other factors. In my dissertation, influence diagrams are used to determine what piece of information in the patient record would have the highest utility for a desired outcome (verifying a diagnosis); this information is then brought to the user's attention by highlighting elements of the UI that provide users with suggestions on what to do next.

2.1.5. Generating explanations

One shortcoming of early expert systems is that they are often a black box. Users input information, and the system outputs an answer. However, the user is never informed of how the system comes to a certain conclusion. An understanding of why the system reached a particular conclusion is arguably just as important if not more important than obtaining the result. An advantage of using BBNs is the ability to generate explanations using properties of the network [20]. Explanations in graphical models have been used to understand how the model performs inference and what assumptions generate the returned outcome. Explanations are useful for: 1) determining what configuration of unobserved variables provides the most probable outcome for the target variable; 2) eliciting what information is contained in a model; and 3) understanding the results of inference on a model and how that particular result was produced. A *static* explanation is generated based solely on the probabilities that are contained in

the model; it does not consider any inputted evidence. On the other hand, a *dynamic* explanation is one that takes into account the evidence inputted to the model. Given a set of evidence variables and a target variable, dynamic explanation tries to find an explanation for any changes in the posterior probability of the target variable due to the introduction of evidence. Dynamic explanations can be further classified as being either micro or macro. A *micro* explanation tries to justify the variations of the probability distribution of a certain node. A *macro* explanation examines the model as a whole and attempts to find the primary paths that link the inputted evidence with the target variable.

Explanations have been presented as natural language statements that provide a textual description of the variables and parameters involved. Druzdzel [21] translates the qualitative and quantitative information of a BBN into linguistic expressions. Probability values are mapped to verbal expressions of uncertainty; for example, the range 0.25 – 0.4 is mapped to the adjective "fairly unlikely." These adjectives are then used in combination with the structure of the network to generate meaningful statements: for instance, given the model depicted in Figure 2.2, one statement would be "smoking commonly causes emphysema."

Visualizing explanations using the DAG have also been explored: [22] utilizes color coding and line thickness to support explanations in terms of weight of evidence and evidence flows. A system that combines both graphical and verbal approaches to explaining inference results is BANTER [23]. It provides an intuitive interface for graph-

ically querying a model and understanding the results: the user first provides a scenario by specifying known values for history and physical findings and the disease of interest using node monitors. The system then uses the model to validate which tests best determine whether the patient has the selected disease. Explanations are generated in natural language by: 1) identifying the evidence that have the greatest impact on a target variable using mutual information; or 2) finding the path between evidence variables and target variable that maximizes the overall impact of evidence variables. Research has also been done to utilize the network topology to aid in the generation of explanations. Yap et al. [24] exploits the property of a Markov blanket to identify a subset of variables that result in a concise explanation of a target variable's behavior. Their approach first restructures the BBN such that the target variable has its Markov nodes as its parents. Then, the target node's conditional probability tables are converted into decision trees. Explanations are generated by traversing the decision trees. Verduijn et al. [25] address the inability of traditional BBNs to provide updated prognostic expectations based on new data that becomes available during the healthcare process. They describe a three-tier system: 1) the first tier is a BBN composed of a collection of local supervised learning models, which are recursively learned from the data; 2) the second tier is a task layer that translates the user's clinical information needs to a query for the network; and 3) the third tier is a presentation layer that aggregates the results of the inferences and presents them to the user using a bar graph. Their novel approach allows users to pose new queries at each stage of patient care (*e.g.*, pre-treatment, treatment, post-treatment), and the model explains the changes in

the target variable based on the updated information at each phase. The variety of approaches described in this section demonstrates the versatility of BBNs to generate explanations using the model structure and parameters; the reader is referred to [26] for additional discussion.

Methods for explanation generation provide a basis for my work in using the graphical disease model to enable context-sensitive visualization. Unlike past work, my approach leverages common visual metaphors used in the electronic medical record as a method of conveying which variables in the underlying model are relevant for a given query. I also translate user interaction with the EMR into a query that can be executed against the model and then use the results to update the display of patient data. Many of the approaches described in the methodology (Chapter 3) are based on past work in explanation generation.

2.2. Other Knowledge Sources

2.2.1. Ontologies

An *ontology* is a formal representation of knowledge that provides a definitive and exhaustive classification of all entities in a domain and their relationships. Ontologies have been widely used to represent biomedical knowledge and have been applied to the areas of knowledge management, data integration, and decision support [27]. They serve as a method for representing a source vocabulary (*i.e.*, list of names for the entities represented). Ontologies can also organize knowledge based on increasingly high-

er levels of abstractions such as the International Classification of Diseases (ICD) and Systematized Nomenclature of Medicine-Clinical Terms (SNOMED-CT). In ICD-9, for example, a brain tumor in the temporal lobe is represented as the leaf node *191.2 - Temporal lobe*, which has the parent *191 - Malignant neoplasm of the brain* and the ancestor *Neoplasms*. Ontologies also enable decision support applications by providing the means to abstract a domain (*e.g.*, brain cancer) and identify unique attributes for that domain (*e.g.*, symptoms, findings, abnormalities). For instance, one application uses the Foundational Model of Anatomy (FMA) to provide anatomical knowledge for predicting the consequences of penetrating traumatic injuries [28]. Knowledge about spatial relations between the path of injury and vital organs is provided by the FMA. I briefly survey three ontologies—UMLS, NCI Thesaurus, and RadLex—which are used in my work to provide domain information for tasks such as query expansion.

2.2.1.1. Unified Medical Language System (UMLS)

The Unified Medical Language System (UMLS) [13] is a hierarchical ontology consisting of two primary sources of semantic information: the Metathesaurus (META), which includes a large, multi-purpose, multi-lingual vocabulary with over 1.5 million unique biomedical and health-related concepts assembled from 150 source vocabularies; and the Semantic Network (SN), which provides a method of categorizing and relating concepts within META using 135 semantic types and related to one another using 54 different semantic relationships. Each META concept is assigned to at least

one semantic type from the SN. The UMLS also includes a SPECIALIST lexicon, which contains general English terms with syntactic, morphological, and orthographic information recorded for each term. In this work, I used the 2008AB release of UMLS.

2.2.1.2. NCI Thesaurus (NCIT)

The NCI Thesaurus [14] is an ontology-like vocabulary that provides coverage of the cancer domain. Terms represented include cancer-related diseases, findings and abnormalities; anatomy; agents, drugs and chemicals; genes and gene products; and others. The NCI Thesaurus combines terminology from numerous cancer research related domains, and provides a way to integrate or link these kinds of information together through semantic relationships. The Thesaurus currently contains nearly 60,000 concepts. In this work, I utilized version 09.02d of the Web Ontology Language (OWL) format of the thesaurus.

2.2.1.3. RadLex

Radiologists have used a variety of terminologies and standards (*e.g.*, American College of Radiology (ACR) Index), but until recently, no single lexicon provided a comprehensive coverage of the field. To address the issue of coverage, RadLex [15] was developed to unify the representation of radiology terms; over 8,000 anatomic and pathologic terms, many of which were not available in other sources. For example, RadLex supplements the anatomic and pathologic codes in the ACR Index with additional types of terms, including: 1) the devices, procedures, and imaging techniques used to acquire radiology images; 2) the perceptual and analytical difficulty of the interpreta-

tion; and 3) the diagnostic quality of the images. RadLex has been applied towards standardizing radiology reporting and defining a set of keywords for indexing and retrieval of images. In this work, I use the frames version of RadLex 2.0.

2.2.1.4. Applications of ontologies

Ontologies have been applied towards helping users organize the available knowledge into semantically related groups: for instance, SemRep [29] uses the UMLS to determine how concepts extracted from a corpus of journal articles on a single topic are related and visually represents their relations using a directed graph. Medical Entities Dictionary (MED) [30], developed at Columbia University, is a semantic network based on UMLS that uses a directed acyclic graph to integrate terms from four hospital systems (laboratory, electrocardiography, medical records coding, and pharmacy). Additional knowledge about each term is added as a semantic link, which assists in the integration and classification of disparate terminologies. MED has been demonstrated as an effective knowledge source for supporting the automatic generation of problem-oriented views of patient data and finding relevant concepts. Other applications have developed visualizations to help users with viewing complex terminology systems. For example, TermViz [31] utilizes a graph visualization package called prefuse [32] that allows users to visually interact with the semantic network. Keywords can be specified to find relevant nodes in the network, called a focus set. Based on this focus set, all other nodes are assigned a degree-of-interest (DOI) value that decreases as the

number of steps from the focus set of nodes increases. This value allows users to adjust how many nodes are displayed based on their proximity to the focus set.

2.2.2. User/task models

User models contain a representation of characteristic information about the user. The model provides a description of the user's information needs and goals; this information can then be used to guide how information is integrated and presented to the user. User models can be characterized along three axes, as discussed in [33]:

- **Canonical vs. individual**. Different models can be formulated for either an entire population of users (*i.e.,* canonical) or individual users: canonical models are generated as part of the implementation of the system and do not change over time while individual models are built and maintained for each new user. In one popular approach, canonical models categorize users into basic groups (*e.g.,* novice, intermediate, expert) called *stereotypes* [34]. Each group has its own unique set of assumptions that guide what interface elements are presented. Individual models adapt to the user's preferences over time by learning how the user interacts with the interface. For example, if the user frequently accesses a particular function or needs a particular piece of information, the interface identifies and changes to make the function easier to perform or to automatically display the information. Many adaptive systems take a combined approach where the default settings are based on a canonical model but as the user interacts with the interface, an individual model is generated.

- **Explicit vs. implicit**. Models may also be classified as explicit or implicit. In *explicit models*, information about the user and task is provided manually by the system designer or user. In *implicit models*, information about the user is collected by the system through the course of normal interaction. Explicit models allow users to customize aspects of the user interface such as changing the layout of the display by dragging and dropping elements on the screen or selecting specific preferences or stereotypes from a list of options. Implicit models try to learn the user's preferences by observing and recording the user's interactions with the system.

- **Short-term vs. long-term**. Short-term characteristics are often associated with preferences or assumptions about a user that are valid over a single session. For example, during one session, a physician may want information about a patient's cardiology reports and electrocardiogram (ECG) results, but during another session, a physician may desire to view other information about the patient's admissions, discharge, and transfer data. Long-term characteristics tend to describe user preferences that do not change across multiple sessions of using the application.

Aside from the user's characteristics and preferences in using a system, there is also a description of the task that the user wants to complete. A task model informs the system of the user's intentions. For example, is a physician reviewing a patient's history through the EMR for the first time, performing a follow-up examination, or documenting an encounter? In each situation, different intellectual and procedural goals are accomplished. A number of task model methodologies have been proposed over the

years [35] to structure task requirements from users. Such models can be used to identify possible usability problems with a system; to assess human performance (*e.g.,* time, efficiency); and to design a user interface [36]. Some commonalities that are shared across different task modeling are noted below:

- **Task hierarchies**. At the core of these models is a way to describe the end objective of the interaction with the system. Rarely are the specifications of tasks atomic: usually several sub-tasks or steps comprise a single task. Thus hierarchical and object-oriented approaches are taken to organize this information, with higher-level abstractions being aggregates of more elementary goals. These frameworks have an implicit constraint that prevents a given task from being completed without all of its sub-tasks being performed.

- **Objects and actions**. Objects are the entities that participate in the task and encompass the user with the resources required to complete the task (*e.g.,* a database, another individual, the system GUI, etc.). Actions are the basic methods that an object is capable of performing.

- **Roles**. A user may change behaviors given different tasks. For instance, a doctor reading a patient medical history may be acting as a physician diagnosing a patient, or may instead be looking at the record as a clinical researcher extracting information. The concept of a role is correlated with that of a user model.

- **Operators and events**. Although task hierarchies provide compositional rules, they do not impose any temporal ordering on (sub-)tasks. Hence, a task model incorporates some mechanism (*i.e.,* operators) that provides relative sequencing between tasks. To describe these constraints, event descriptions are embedded within the model, specifying milestones and/or conditional dependencies.

2.2.3. Evidence-based medical guidelines

Clinical guidelines provide a method for standardizing how medical care is administered to a patient; it specifies a set of plans at varying levels of abstraction for screening, diagnosing, or managing patients who have a particular medical problem, need, or condition. According to the Institute of Medicine, clinical guidelines are, "systematically developed statements to assist practitioner and patient decisions about appropriate health care for specific clinical circumstances" [37]. Guidelines can be used to generate reminders or alerts for a single time point when the patient meets a certain criterion or for long-term plans that specify how patients should be treated over an entire course of an illness. They have been demonstrated to improve the quality of medical care and in some cases, the overall survival of patients [38]. While not explicitly used in my work, they provide an alternative source of context that may be used to transform task models. However, one limitation is that they are currently only available for common diseases.

Most clinical guidelines are text-based and not easily accessible to care providers, who need to match them to their patients and to apply them at the point of care [39]. This

problem is compounded by the fact that computers have no means for utilizing this knowledge because the representation is not machine comprehensible. Efforts such as ONCOCIN [40], PROforma [41], GEM [42], and GLIF [43] are examples of structuring guidelines into a uniform computer-interpretable language such as extensible markup language (XML). Hence, there is a need for automating the integration of guidelines into the existing infrastructure for presenting patient information to physicians. Here, we examine how PROforma may be used to provide context for displaying medical data.

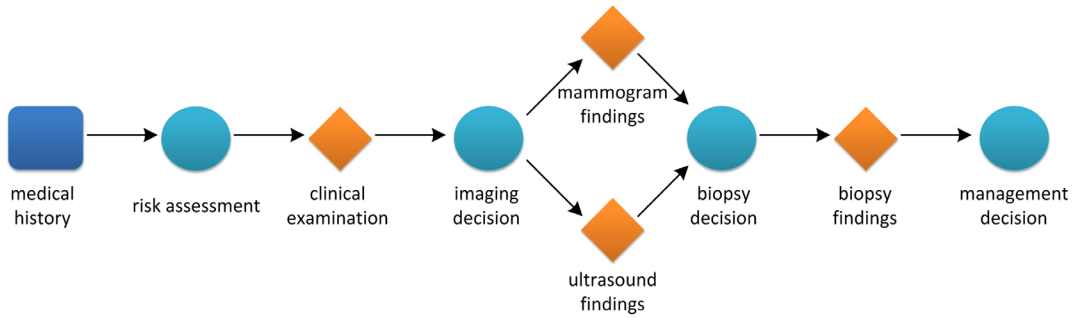


Figure 2.7: Example of a workflow for managing patients with symptoms of breast cancer represented in the Triple Assessment and Diagnosis System [3]. The decision nodes are represented by circles and embedded at various points in the workflow. Diamonds represent points where patient data are acquired.

PROforma. The PROforma language [41, 44] is represented as a type of graphical model: nodes represent concepts and the edges represent inference processes. Nodes collectively represent a task, which can individually or collectively represent a clinical guideline. Four types of nodes (representing a different type of task) can be defined: a *plan*, which is a container for other tasks with an ordering imposed to reflect temporal,

logical, resource, or other constraints; a *decision*, which is a set of argument rules determining the options that should be chosen according to current data values; an *action*, which is a procedure linked to a process that can be executed; and an *enquiry*, which is a request for more information that is needed to complete a procedure or make a decision. Each task can take on a number of properties; properties can be a scalar value, an expression, or an object that contains additional properties. All tasks and data items are associated with: a description, which provides an explanation of the intended purpose of a task; a precondition, which is a truth-valued expression that is true when a task is started; and task scheduling constraints, which are logical constraints that prevent one task from starting before another task.

Creating a PROforma application is a two step process: 1) a high level description using a graphical model is created that depicts the clinical tasks along with their logical and temporal interrelationships; and 2) details (values) for each clinical task are defined. An application of PROforma in the clinical environment is Triple Assessment and Diagnosis System (TADS), which is used to diagnose and assess the risk of women with symptoms of breast cancer [3]. The general workflow of TADS is depicted in Figure 2.7. The guideline utilizes three types of information to come to a decision: 1) clinical examination; 2) imaging (*e.g.*, mammography, ultrasound, MRI); and 3) histopathology (*e.g.*, fine needle aspiration). The decision node decides whether the patient should be discharged or referred to a team for follow-up and treatment. The structure

of PROforma facilitates translation of its information into a task model that can be used to select relevant data elements for inclusion in a display.

2.3. Medical Visualization

Visualization is broadly defined as, "the act or process of interpreting in visual terms or of putting into visible form." Information visualization is more specifically defined as the process of "transforming data, information, and knowledge into visual form making use of humans' natural visual capabilities" [45]. The goal of information visualization in medicine is to help physicians: 1) explore available data at various levels of abstraction; 2) engage the user through interactive techniques; 3) encourage discovery of details and relations in the data that would be difficult to notice otherwise; and 4) support the recognition of relevant patterns by exploiting the visual recognition capabilities of users [45]. Representations such as sparklines, Gantt charts, treemaps, and spiral graphs are a small sampling of the many information visualizations that have been used. This section presents a brief review of common visualizations being used in medicine; presented visualizations are organized based on the taxonomy proposed by Starren [16]. Visualizations are categorized based on the type of representation: the lowest level of sophistication involves the presentation of textual and numerical data in a relatively non-interpreted fashion; lists and tables represent this category of presentation. Next, for quantitative and statistical information, understanding the data involves comparisons and trending: visually, different types of plots and charts are used to emphasize relative values, with the manner of such displays largely influencing in-

terpretation. Expounding more conceptual relations, graphs and trees provide a further degree of abstraction. Finally, the topmost level of this hierarchy comprises the visual abstractions brought about through pictograms, which collectively aim to be graphical surrogates for real-world entities and concepts seen in the clinical environment [33].

2.3.1. Lists and tables

Text and numerical data are the predominant component of the patient record. The most familiar method of displaying sequences of related information, *lists* are enumerated or delineated sets of textual and/or numerical items. Typically, the entries in a list are short and concise, presenting a key point or summary that can be quickly read by the user. Examples from clinical practice include an individual's medical problem list; physician worklists (*e.g.,* imaging studies awaiting interpretation); and a patient's set of current medications. Aside from a straightforward display of list items, today's GUIs show lists in a number of different ways, imposing different modes of interaction and selection. For example, combination boxes (combo boxes) enforce selection of a single item, while checkboxes allow for multiple items from a group of related entries. List entries can serve as hyperlinks, allowing a user to access further information. Lists are generally univariate in that a single concept is being communicated per item. *Tables* (also referred to as *grids*) can be seen as extension of lists to present multivariate information, where each row in a table in a single entity, and each column is an attribute of the entity. An archetypal use of tabular views in medicine is the comparison of different lab panel values over a set of dates in flowsheets. Adaptations on

tables include colorization and re-orderable matrices. In the first variant, the range of values for a variable is mapped to a color spectrum so that cells are filled with a color rather than a number. The second variant enables the rows and columns to be sorted or arbitrarily arranged to facilitate pattern discovery. *Heatmaps* use both colorization and re-ordering [46] and are widely used to visualize large quantities of data such as in the analysis of expression data from DNA microarray hybridization experiments.

2.3.2. Plots and charts

Information presented within tables, although precise, fail to foster rapid interpretation of subtle trends, especially over a large number of data points. Thus, the next level of graphical abstraction seen with medical data involves *plots* and *charts* (the terms being used interchangeably), wherein the relative nature of numerical data is contrasted to illustrate changes in values or comparative differences. Data presented in tables can be transformed into a suitable chart to visually accentuate patterns. Elementary graphical charts include:

- **Line and scatter plots**. A common graphical representation is the 2D *line plot*, wherein one axis (*e.g.,* the *y*-axis) represents the quantitative value of interest (the dependent variable), and the second axis (*e.g.,* the *x*-axis) is the space over which the value is sampled (the independent variable, *e.g.,* time). For instance, an ECG is representative of a line plot, where the amplitude of an electrical signal is charted over time. Likewise, a given laboratory value (*e.g.,* blood glucose) may be plotted to visualize increasing/decreasing trends. *Scatter plots* are a generalization of the

line plot, often used in research studies to find potential associations/correlations between two variables over a population; again, one variable is explanatory or controlled, and the second variable is the response or observation. Dimensional scaling techniques (*e.g.,* principal component analysis, PCA) can be used to reduce the number of attributes involved, thereby mapping a multivariate visualization problem to 2D where patterns may be more evident. If no association exists between the variables, no discernible visual pattern or trend is seen in the scatter plot.

- **Bar charts and histograms**. Another well-recognized type of plot is the 2D *bar chart*, where the length of a rectangle is used to proportionally depict the value of a given category; multiple categories are then compared. Additionally, parallel comparisons between datasets can be visualized in a bar chart, facilitating intra-category comparison. To demonstrate, in a clinical trial for a drug a bar chart may be used to show side effects (*i.e.,* categories) with the percent of individuals affected. Parallel bars may then be placed adjacent to compare these individuals versus a control group (*e.g.,* placebo). *Histograms* are a specific type of statistical bar chart, wherein the categories represent tabulated frequencies of a given value (or values over uniformly divided ranges). An image histogram, which plots the number of pixels with a given intensity value, is representative of this information graphic. For histograms, the choice of discretization can greatly change the understanding of the data. While variations of bar charts exist employing different graphical techniques (*e.g.,* 3D bar charts, stacked bar charts), the overall complexi-

ty of these presentations and a user's ability to correctly interpret the data can outweigh their utility.

- **Pie charts**. A *pie chart* aims to provide a sense of proportion by dividing a circle into wedges, representing an object and its constituent breakdown. One exception to the pie chart paradigm is a more complex variation, the *polar area diagram*: rather than use the angle of a wedge to convey percentage, the angles of each wedge are equal and the radius varies in proportion to the amount. The end effect of a polar area diagram is that the pieces project outwards, making similar quantities easier to relate and compare.

- **Radar charts**. Less widespread are *radar charts* (also called *circular* or *spider charts*), which compare three or more quantitative variables along multiple axes. The axes radiate outwards from the center of the plot, along which the data values for each variable are drawn on a shared scale. However, variants of radar charts have been defined to take advantage of shape and area by connecting the plotted points. [47] introduced this concept for clinical labs, with normalized values for laboratory data charted as a shape. Ideally, if the lab values are balanced, the shape will conform to the overall geometry of the radar plot (*e.g.,* for a lab panel with six tests, the overall shape should resemble an equisided hexagon); in contrast, skewed labs distort the overall shape, allowing the viewer to quickly identify which axis (*i.e.,* lab) is discrepant and the direction of imbalance (low values gravitating towards the center of the plot, high values being on the edge). Adaptations of the ra-

dar graph also use area to compare different observational sets (*e.g.,* two time points): the overlap of an area and trends can be seen.

2.3.3. Graphs and trees

Plots are intended to express numerical data; in contrast, *graphs* and *trees* are designed to demonstrate relations between concepts. In this section, the terms "graph" and "tree" refer to the formal constructs defined in computer science, as opposed to more generic pictorial constructs.

Apart from their use in evidence-based medical guidelines as *flowcharts* illustrating decision pathways (*e.g.,* eligibility criteria for a clinical trial, study design), graphs are generally not seen in clinical practice. Outside of the clinical arena, conceptual graphs and (probabilistic) graphical models have a longstanding history within medical informatics, being used to represent ontologies and as a part of decision-support frameworks (*e.g.,* Bayesian belief networks (BBNs), hidden Markov Models, Petri nets, etc.).

Trees are used to illustrate connections between entities where the entire structure of a hierarchy and its encompassing relations are relevant: parent-child relationships (*e.g.,* is-a inheritance); siblings (objects at the same level in the hierarchy); and clusters are visually portrayed, usually with the root of the tree being the most general concept and levels further out (*i.e.,* toward the leaves) becoming more specialized. Information arranged as nested lists are amenable to tree presentations; hence, controlled vocabularies and clinical coding schemes are often shown as trees [48, 49]. Other clinical ex-

amples of trees include: grouped medical problem lists (*e.g.,* symptoms and diseases by anatomical region); composite lab tests (*e.g.,* a metabolic panel); an imaging study and its constituent series; and structured reports, wherein a document section may consist of multiple sub-parts. *Dendrograms* are a specific type of graphical tree used to envisage related groups; taxonomies and genomic analyses involving hierarchical clustering algorithms are indicative of this graphical element.

2.3.4. Pictograms

The highest level of visual conceptualization comes about in considering the use of *pictograms* to represent clinical concepts. A pictogram is defined as a graphical symbol that represents a concept or entity. The use of pictograms with medical data can be seen in four different areas:

- **Icons**. *Icons* are small pictograms, and are a familiar component of modern GUIs representing an action or data object. Pictograms can be linked to descriptive phrases and the numerical scale; one example uses a variant of the visual analogue scale (VAS) to present a patient with a spectrum of facial expressions as part of a health questionnaire. The goal is to assist patients with describing their level of discomfort. One limitation is that the interpretation of icons can be subject to personal and cultural biases [50, 51], thus making the use of icons across populations complex. In certain cases, the graphic is universally understood [52]; but in domain-specific scenarios, individuals may initially need assistance in understanding the suggested visual cue [53]. Icons can also be derived from an object's content.

For instance, TileBar [54] takes in a set of documents and user-specified terms to generate multiple small bars that are divided into smaller, color-coded squares. Each square represents individual terms and indicates relative document length, query term frequency, and query term distribution. TileBar thus provides a quick pictorial representation of document content and relevance to keywords (*e.g.,* disease name).

- **Maps**. *Maps* are larger pictograms, being mainly concerned with a spatial framework (*e.g.,* an anatomical atlas). For instance, maps are used as surgical drawings to document the planned approach, and the pre- and post-operative state of the region of interest. Whole-body anatomical drawings are also used to quickly demonstrate affected or symptomatic areas. Maps can also used to represent high dimensional data, such as the contents of a clinical report. [55] abstracts a text document into a Kohonen's feature map using visual cues such as dots, clusters, and spatially-related areas to represent the unique concepts (*e.g.,* disease, drugs, chemotherapy), the frequencies of word occurrence in titles and frequency of word co-occurrence respectively.

- **Diagrams**. *Diagrams* are illustrated figures that present an abstraction or conceptual metaphor (*e.g.,* a timeline for a therapeutic regimen; a structural depiction of a medical device). Although a clinical diagram may also be anatomically based, the primary difference between a map and a diagram is the intended communication of a spatial vs. non-spatial relationship, respectively.

- **Images**. Lastly, medical *images* are a physical representation of the real-world based on either light (*e.g.,* optical photography, such as seen with dermatology and post-operative surgical procedures); radiation (*e.g.,* CT, nuclear medicine); or other physical value (*e.g.,* hydrogen nuclei interaction/relaxation, such as under magnetic resonance). The rendering can be a 2D projectional or cross-sectional image, showing spatial relationships (*e.g.,* between a tumor and normal tissue); a 3D reconstruction; or a 4D representation (*e.g.,* an animated 3D visualization showing changes over time).

The above categorization of clinical visualizations is only intended to provide insight into some widespread graphical elements used to communicate clinical concepts and data: it is by no means comprehensive. Several of these visualizations have been implemented in my work to visualize different data elements that exist in the patient record; the process of matching data with visual metaphors is described in Section 3.5.2.2.

2.3.5. Interaction

Users need tools to interact with visualizations to uncover new insights by posing queries to the data and identifying patterns in the results. The variety of interactive methods have been organized into taxonomies (*e.g.*, organized by low-level techniques [56], interactive techniques [57]). Here, techniques are categorized into groups based on a combination of user objectives and the interaction techniques that accomplish them [33]:

- **Selecting**. The act of selection uniquely identifies a single data point by highlighting it (*e.g.,* using a different color) so that users may visually track the location of items of interest. Typically, selection occurs as the first step of a series of interaction techniques as a method to identify a subset of data elements that the user is interested in exploring further.

- **Exploring**. The amount of information displayed is limited by the screen size and the user's ability to perceive an array of presented information simultaneously. If the amount of data is too much to fit into a single screen, tools are needed to explore the data. Actions, such as panning and scrolling, allow users to intuitively move data across the screen and configure the display to show data of interest.

- **Reconfiguring**. Sometimes a single perspective of the data is insufficient to fully understand any patterns or trends. Reconfiguring the dataset allows users to change how the data is presented by viewing the same data in different arrangements. For instance, in multidimensional scatter plots, new views of the data are generated by changing the attributes presented on the axes.

- **Encoding**. Compared to reconfiguring, encoding allows users to transform the representation of a data element from one form to another. For example, a pie chart may be a more effective display for a particular dataset than a histogram. Encoding may also involve reassigning visual attributes (*e.g.*, color, size, shape) to better differentiate clusters of data.

- **Abstracting**. Data may be viewed at varying levels of abstraction. A common technique is allowing users to *zoom* between broader and more detailed views. An overview may be used to obtain a general idea of the data; however, users will want to magnify specific regions in the data that is of interest to them and view additional information.

- **Filtering**. When displaying large amounts of data simultaneously, users need tools to help identify and focus on the data relevant to their task. Filtering is a technique that allows users to conditionally hide or change the appearance of certain data points that do not fall within specified criteria. If a physician is examining a patient who has hypercholesterolemia, unrelated documents should be filtered (*e.g.,* reports on a broken leg), leaving only a subset of documents pertinent to the treatment of high cholesterol.

- **Connecting**. When multiple different visualizations are used to represent the same data, the correspondence between each view may be highlighted by linking them together. For instance, if a user selects a set of data points in one view, all of the views reflect the same selection in their own way. This process is called *brushing*.

The aforementioned methods of interaction are typically used in combination to provide users with graphical tools for manipulating data and posing queries. Traditionally, users interact with a database by formulating textual queries using machine-understandable languages such as structured query language (SQL), which features a non-intuitive and difficult syntax for non-programmers to learn. To address these is-

sues, two categories of querying frameworks have been used in medicine: direct manipulation and query-by-example. Both frameworks share several characteristics: 1) they provide graphical representations of real-world objects and actions; 2) they use a pointer to select or identify an element; 3) they allow rapid, incremental, and reversible actions to be performed on the data; and 4) they provide immediate and continuous display of results. In particular, direct manipulation principles have been shown to assist users with navigating large information spaces [58].

Direct manipulation. Direct manipulation interfaces model how people interact with objects in the real world by providing users with tools to interact with visual objects that represent the data elements [59]. Several benefits exist for applying direct manipulation to data querying: 1) the user does not need to learn a complex query language in order to pose a valid query to the system; 2) the user does not need to worry about making syntax errors; and 3) the user obtains immediate feedback about the query posed and potential size of the results [60]. Many applications have been developed using diagrammatic visual data querying [60, 61]; a select few are reviewed here. ADVIZOR [62] is a commercial system that works with data cubes to query aggregated data. The system is implemented using a relational model. Users select data using either a tool that makes predefined (*e.g.*, rectangle, circle) or freeform shapes. The system allows new selection sets to be related with an existing set by using expressions such as replace, add, and subtract. When an event occurs in a view, that view notifies the corresponding data table, which in turn updates its state. The data table

then notifies all visualizations using this data to update their representations. Another system is IVEE/Spotfire [63], which automatically creates a dynamic query application from a given database schema. It has a collection of visualizations (*e.g.*, histograms, bar charts, pie charts) that are selected based on the attribute data types within the application's data schema. Each visualization is tightly linked so that changes to one view affect how the others are presented. In the medical domain, TraumaSCAN [64] utilizes a three-dimensional model with which users interact to place entry and exit wounds for injuries from gunshots. In combination with inputted patient findings, the system performs reasoning on a Bayesian model to predict the most probable symptoms and conditions arising from the specified injuries.

Query-by-example. This interaction paradigm asks a user to sketch or provide an example object as the basis of a query and attempts to find all objects in the database with similar visual attributes. Seminal work in this area include systems developed by Chang [65] and Joseph [66]. Chang presents a relational query language introduced to simplify the usage and management of image data. Joseph discusses a set of manipulation operations that should be supported by an underlying pictorial database management system and a higher-level query language that carries out these operations. Dionisio [67] describes a visual query language called MQuery, which uses a single set of related query constructs to interact with data stored as time-based streams. The work presents a visual interface to interact with these streams in the context of querying multimedia, timeline, and simulation data. Del Bimbo [68] presents methods for

translating sketches of object shapes into queries that find indexed icons; a similar system, DOODLE [69], facilitates searching a database using pictures. These works lay the foundation for future applications to support pictorial querying. One such application is geographical information systems (GIS), namely because geographic concepts are often vague, imprecise, little understood, and not standardized. Egenhofer et al. [70] discuss work on a spatial query-by-sketch system that automatically translates the spatial layout of query objects into a database-understandable query. Sketching provides an intuitive way for users to express the image-like representation of spatial configurations that are in their minds. In medicine, visual querying has primarily been used in content-based image retrieval. Abate et al. [71] illustrate a system that allows a physician to query a collection of thoracic images using a combination of visual metaphors. By identifying abnormalities in an image using edge detection, the system then extracts spatial locations, opacities, shapes, and geometrical measures of the abnormality and compares them against features indexed in a database. Sasso et al. [72] present a similar system that provides a user with tools to query a thoracic imaging database using a combination of template and user-drawn features. They evaluate their system using eight query images; for each query, they compare the system's results with those selected by a group of radiologists. They found their system to have a 97.5% and 91% recall and precision rates, respectively.

2.4. Integrated Display

Researchers have long attempted to create a longitudinal, virtual patient record that seamlessly incorporates data elements from the patient record into a single display [73]. The NUCLEUS project [74] generates a customized, integrated multimedia record based on templates that dictate how data sources are hyperlinked from a single display. Puya [75] reduces the presented information by filtering out sentences in the medical narrative that refer to normal findings leaving only abnormal ones for the physician to see. QCIS [76] creates multiple views of the same data based on where the data is acquired (source-oriented), when the data was acquired (time-oriented), or how related the information is to a given medical problem (concept-oriented). Effort has also been directed towards temporal visualization of patient records; the principle of using timelines was first introduced by Weed [77] and then initially represented in graphical form by Cousins [78]. LifeLines [79] is a popular example of this type of display; in addition to using timelines to display chronologies of medical data, it color-codes each event based on category (*e.g.*, disease, laboratory value) and groups data elements based on type (*e.g.*, notes, tests). LifeLines2 [80] adds a set of operators that allow users to dynamically reorganize the presentation of data based on a certain feature. For instance, all of a patient's events (*e.g.,* physician encounters, high blood pressure) may be aligned based on the proximity to an event of interest (*e.g.,* heart attack). Bui et al. [81] have developed TimeLine to reorganize patient data based on medical problems and dynamically render relevant data elements on a timeline. The novelty of

their work is its ability to filter data elements that are displayed so that only ones that are related to the selected disease appear on the timeline. Filtering in this work is accomplished using a predefined set of rules.

[82] has shown that radiologists are increasingly in need of complete access to patient record in order to perform the appropriate diagnostic service. Diagnostic protocols are highly dependent on identifying trends and relevant events within the patient's medical history and prior findings. The interpretation of medical images also requires a better appreciation of how information from other parts of the patient record interplays with these findings. With the complexity and improved diagnostic performance of modern imaging techniques, being able to understand clinical context is essential for adequate image interpretation [83]. Hence, seamless integration of data from multiple information systems into a single display is critical.

2.4.1. Organization of patient records

The role of the modern patient record is multifaceted: to support patient care by providing a source of evidence for evaluation and decision making and for communicating among different care providers responsible for an individual; to be a legal record of medical actions; to support research; to educate physicians; and to streamline healthcare administration. As such, multiple methods for organizing the entire patient record has been proposed:

- **Time-oriented.** In a time-oriented organization, all clinical observations are recorded in chronological order. While making the record easy to maintain, clinicians

experience difficulty associating disease episodes that are scattered over time. This problem is compounded as the number of problems in the patient record increases: this organization provides no clear way of filtering or rearranging the data to show only data elements that pertain to a specific disease.

- **Problem-oriented.** As first presented by Weed [84], a problem-oriented medical record utilizes each disease that is documented in the patient's record as a classifier, and all documents, labs, and images are clustered together based on whether they are related to that particular medical problem. For a patient with liver disease and a lung cancer, his data would be clustered around these two problems. Data elements such as CT images of the lung, pulmonary function test results, and lung needle biopsy results would be grouped under "lung neoplasms". However, maintaining a problem-oriented record is time intensive and may result in the creation of redundant data.

- **Source-oriented.** In the source-oriented organization, data is sorted chronologically by how they were obtained (*e.g.*, department). For instance, all of the patient's radiology reports are organized separately from his cardiology reports. While this organization may improve a clinician's ability to quickly view reports generated by a particular department, it has the issue of fragmenting patient information across several sections.

2.4.2. Existing systems

Earlier, we discussed different approaches that researchers have presented to address the problem of integrating patient data in a single view. It would be useful to describe the current state of medical record visualization by examining two systems that have been deployed in a patient care setting: the Veterans Health Administration's Veterans Health Information Systems and Technology Architecture (VistA) and University of California, Los Angeles' Patient-centric Information Management System (PCIMS).

VistA. VistA was introduced in 1996 to support the day-to-day operations at local Veteran Affairs (VA) health care facilities. The system is built on a client-server architecture, tying together workstations and personal computers with graphical user interfaces. VistA is one of the world's largest implementation of the EHR to date with over 4 million veterans' records in the system accessed by 180,000 medical personnel across 168 different sites. In addition, some private or community health networks have deployed an open-source version called OpenVista in their clinics. The goal of VistA is to provide a single interface for health care providers to review and update a patient's medical record, and place orders such as medications, procedures, imaging studies, and laboratory tests. The interface is designed to display all information that supports medical decision making; a summary screen (Figure 2.8, top) provides timely information about the patient's active problems, allergies, current medications, recent laboratory results, vital signs, hospitalization record, and outpatient clinical history. For instance, when a clinical document is selected (Figure 2.8, bottom), information is

displayed immediately with relevant images and attachments displayed along with the body of the report. The strengths of VistA are: 1) the system is scalable and provides integrated access to the multimedia patient record; 2) information is organized by source and time; and 3) physicians have access to graphical views of laboratory results. However, the system also has several limitations: 1) information is not filterable by problem; 2) the user interface is static for all users despite their different information needs; and 3) the system does not incorporate any type of decision support tools.

PCIMS. At UCLA, patient records are digitized and accessible through a secure web portal called PCIMS. The interface allows physicians to access a variety of textual data such as demographics, record of past encounters, clinical notes and documents, laboratory test results, ECG reports, operating room cases, and imaging studies. This information is integrated by a horizontal navigation menu displayed at the top of the screen that links out to individual resources (Figure 2.9a). PCIMS is an example of a source- and time-oriented patient record. After specifying a specific data type (*e.g.*, documents) to view, a listing of all available data elements for that type appears; if the user selects documents, the entire list of patient documents appear, sortable by date, title, signing author, and type. The left-sided menu provides options for filtering subsets of the documents by criteria such as "all documents without notes", "progress notes", and "oncology notes" (Figure 2.9b). If the user is interested in viewing laboratory test results, a separate page is loaded displaying all of the values using formatted text. Basic functionality is provided to filter results by group (*e.g.*, metabolic panel),

and results are annotated with a '@' symbol to denote any abnormal values. In the current interface, users need an understanding of what type of information each source provides so that they can locate the appropriate document. The limitations of this design are that: 1) the results are presented in text so any temporal trends and meaningful patterns are not evident; 2) documents can only be sorted by certain fields (*e.g.*, timestamp, data type) and not by semantic concepts (*e.g.*, medical problem); and 3) users need to view laboratory, report, and imaging data individually. Features from each system are summarized in Table 2.1 and compared to one application that has been developed as a result of this work, AdaptEHR (Section 4.3).

2.4.3. Context-sensitive visualization

Up to this point, I have described the various sources of domain knowledge that are available and the different visualizations capable of representing medical data. However, these discussions have not yet addressed the problem of using the domain knowledge to adapt the user interface to a unique user's needs. [85] finds that simply using a graphical technique to render all the data does not provide adequate support for a user's task. Indeed, the user's expectations and preferences play an integral part of the visualization process. Visualizations for large or complex datasets need to be aware of the context in which the data is being used and adjust the presentation to match the user's preferences. An *adaptive interface* (or adaptive hypermedia) [86, 87] conforms to a particular user's preferences by utilizing a model based on the user's past experience with the system and uses this information to dynamically adjust the presentation

of data to conform to a user's individual preferences. Debevc's work [88] presents an adaptive toolbar that offers suggestions for adding or removing command icons, based on the frequency and probability of specific commands. The system uses a decision engine that takes into account the overall frequency of use for a particular command, recent frequency, successive frequency, and overall pattern of use over the entire period. The decision engine decides whether a command icon is relevant solely on the number of times a user interacts with it; therefore, it can only passively suggest commands based on previous use, rather than suggest commands that may be of use in real-time. Horvitz et al. [89] describe the construction of Bayesian models for reasoning about the time-varying goals of computer users from their observed actions and queries. The approach has been implemented in software such as Microsoft Office (*e.g.*, office assistants) with limited success: while the system attempts to predict what functionality is useful to any user, in reality, suggestions are often misguided because the variation among users is too large. The system might have been more successful if it were customized for a specific subset of users. In the medical domain, Mars Medical Assistant [90] utilizes a combination of user, situation, and task to make automated suggestions about related topics that are of interest to a physician examining a particular disease. Cimino et al. [91] have developed a similar but more widely used system called an "Infobutton Manager", which automatically provides links to related information based on the information that the user is currently viewing. Adaptive interfaces continue to be an active area of research as applications become more complex and interfaces become more cluttered.

The idea of context-sensitive visualization is an evolution of adaptive interfaces. While adaptive interfaces focus on changing portions of the interface (*e.g.*, toolbar, content pane), context-sensitive visualization is capable of changing not only the appearance of the user interface by altering the visualizations used to compose the display but also the content that is rendered by these visualizations. In medicine, different users can have unique information needs: a nurse may be interested at gauging the patient's immediate vital signs while a primary care physician is more interested in the patient's long-term outcome. Even within a single user group (*e.g.*, physicians), information needs can vary based on task. Among physicians, for example, tasks can be summarized by three goals [92]: 1) they wish to become familiar with a new patient; 2) they are looking for specific details; or 3) they are searching for evidence to support or discount a hypothesis. However, given time constraints and the desire to increase efficiency in the healthcare workflow, clinicians do not have the time to examine the entire patient record to complete these tasks. Rather, physicians often access information efficiently and specifically, oftentimes only briefly looking at patient history to obtain context but focusing much of their time on specific data elements such as medications, vital signs, and laboratory tests. These studies demonstrate the importance of customizing the display of information for individual users and tasks. In the next chapter, I show how the BBN, combined with other knowledge bases (*e.g.*, user/task models), provide a mechanism for adapting the display.

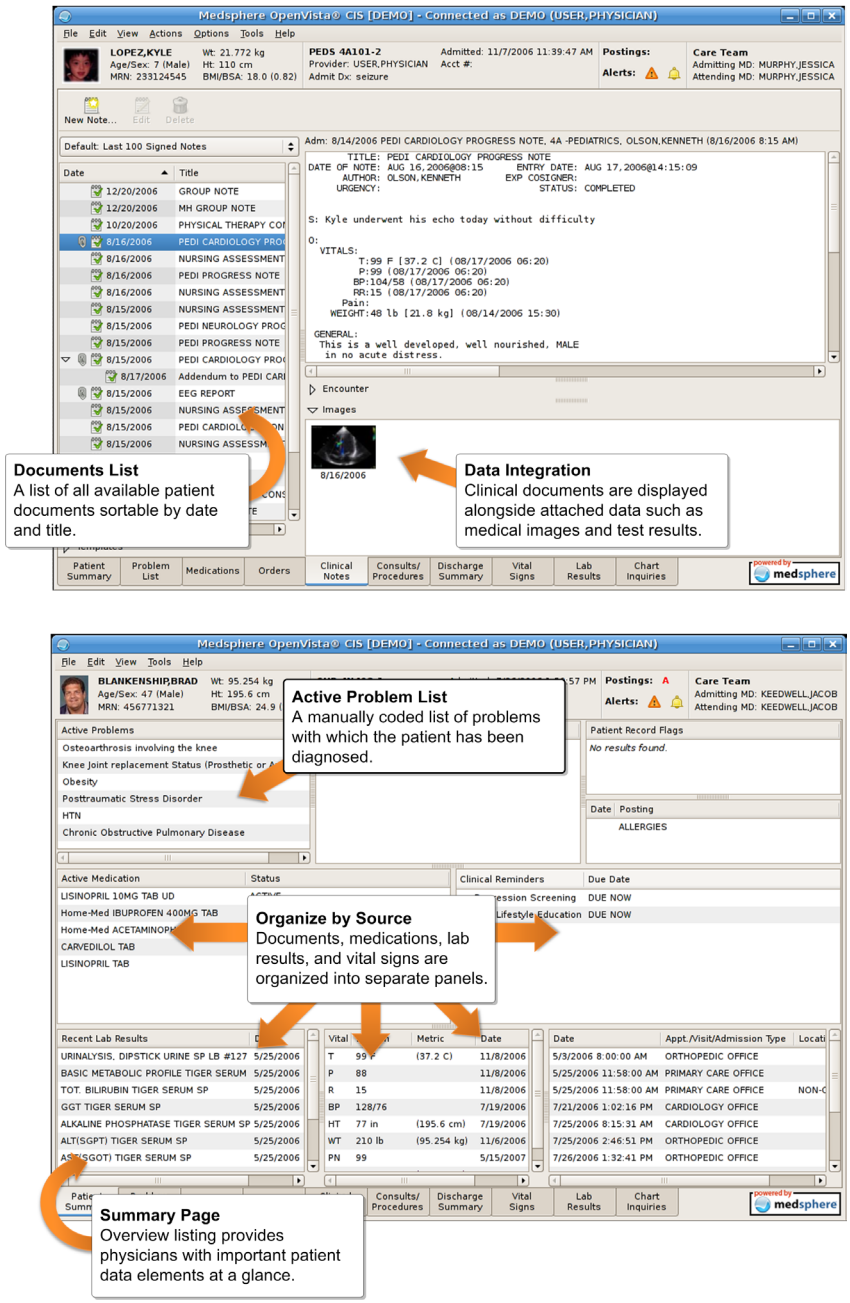
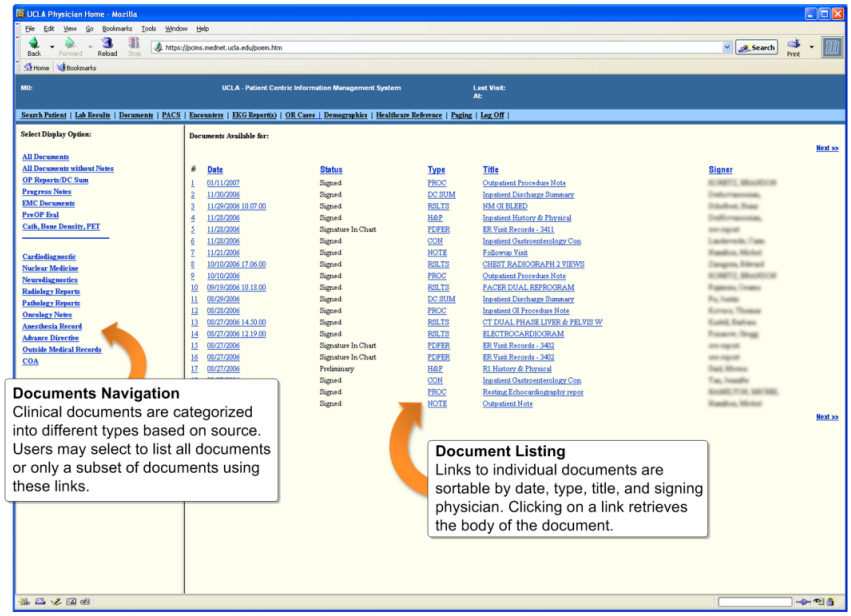
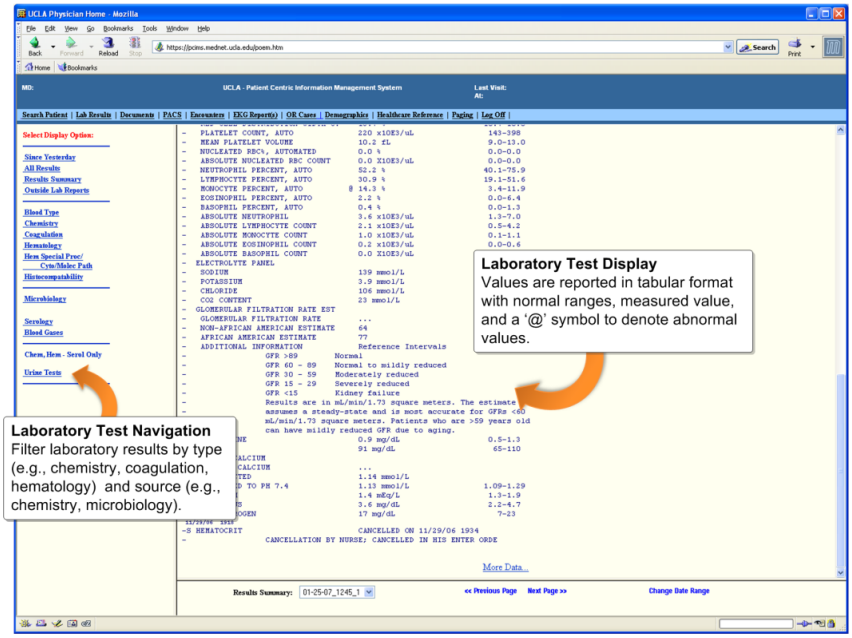


Figure 2.8: Screenshots of OpenVista (courtesy of Medsphere, http://medsphere.org), an open source version of VistA, which is a health information system developed by the Department of Veterans Affairs. The screenshots highlight various features of how patient data is organized and presented to a clinician.



(a)



(b)

Figure 2.9: Screenshots of the UCLA PCIMS system: **(a)** Listing available documents in the patient record. **(b)** Showing laboratory results.

Feature	VistA	PCIMS	AdaptEHR
Display clinical documents	A simple document viewer is supplied with ability to open any attachments (*e.g.*, ECG results, imaging study)	A single web page with the document formatted as either text or Adobe Acrobat file	A document viewer that highlights key concepts and displays associated data (images, medications) in a split panel
Display laboratory values	Options to view both tabular and graphical views of lab results	Only a tabular view is shown. A '@' symbol is used to denote abnormal values	Lab values can be displayed using a table, chart, radar plot, or a combination of the three
Graphically highlight trends, abnormal values	A graphical plot of with color-coded points to denote any abnormal values	N/A	Abnormal points are highlighted in red; user may hover over individual points to obtain exact value
Display medical images	View images alongside document as attachment	Links to a separate radiology application	Users can view images as either a thumbnail, detail-in-context view, or detailed view with a fully integrated image viewer application
View encounter data	Listed in patient summary page	Listed in a separate web page	N/A
Order medications/tests	Yes	Yes	N/A
Generate a summary of patient data	Summary page incorporates access to many common data elements in one screen	N/A	A timeline display is generated that plots all of the different types of data available and ranked by user-defined filters
Rank documents based on relevance to a medical problem	N/A	N/A	Graphical disease model is used to rank which documents are most relevant for a given context
Use patient information to query decision support system	N/A	N/A	The disease model can answer diagnostic, prognostic, and therapeutic queries
Ability to search record by keyword	Search by common fields	N/A	The interface provides a search field for querying data elements
Ability to filter/rearrange patient data	Provides options to search and sort data elements by various fields	Has the ability to sort listings by basic headings (*e.g.*, name, title, date)	Has the ability to create user profiles that adapt information based on individual user preferences

Table 2.1: A comparison between existing electronic medical record systems and the system developed as a result of this dissertation, AdaptEHR. N/A denotes that the item is not applicable/not available.

CHAPTER 3

Methodology

3. Overview

The crux of context-sensitive visualization is customizing the application's GUI to adapt to the user's needs in changing contexts of use. This chapter discusses methods that are used to obtain context by examining the variables, structures, and parameters of a graphical disease model: knowledge from the disease model is combined with other sources of context, such as user and task models, to influence how information is presented in the GUI by varying visual attributes such as position, size, layering, and opacity. The first part of the chapter (Sections 3.2-3.4) addresses four problems: 1) how variable names can be used to link variables in the disease model to elements in the patient record; 2) how the structure and parameters of the model can be used to quantify relatedness and strength of influence between variables; 3) how information from queries can be used to discern user needs; and 4) how other knowledge sources can be utilized to impose additional constraints on what information is displayed. The last part (Sections 3.5) explores how context obtained from the graphical model is translated into a set of composition rules that are used to visually combine, highlight, and lay out specific elements of the patient record. The visual dictionary is presented as a data model for incorporating disparate sources of context.

3.1. Example Queries

The goal of developing context-sensitive visualizations is to assist users with answering a wide range of clinical questions using the available patient data. In this section, I provide several example queries as a basis for discussing how the adaptive display of patient data works. The purpose of posing these questions is threefold: 1) the queries demonstrate the wide range of questions that may be executed against a graphical disease model; 2) they illustrate how different users and tasks have diverse information needs; and 3) they provide a starting point for illustrating the differences in how patient data is rendered based on context.

Query 1 What is the most probable range of the Karnofsky Performance score (KPS) for a 50-60 year old female with a right occipital lobe GBM immediately following complete surgical resection?

Query 2 What is the expected time to progression (TTP) at time of diagnosis for a > 60 year old male patient with GBM in the left temporal lobe if he is on the chemotherapy drug temozolomide?

Query 3 Given that the patient has elevated total bilirubin and γ-glutamyl transpeptidase (GGTP) levels, what additional information is needed to conclude that the patient has primary biliary cirrhosis (PBC)?

Query 4 What document in the patient record provides the most information about the patient's treatment history with respect to his glioblastoma multiforme?

Query 5 Display all available data of interest to a radiologist collected during the same time period that the patient was on gemcitabine.

Query 6 Which patient cases that are present in the database are similar to one with a lower left lobe stage III non-small cell lung cancer who has an Eastern Cooperative Oncology Group (ECOG) score of zero?

Query 7 Has the patient experienced any abnormal laboratory values after being prescribed dexamethasone?

Query 8 Is my non-small cell lung cancer patient eligible to participate in a study of that compares erlotinib to standard chemotherapy?

3.1.1.1. Query processing: Overview

Here, I examine how each query might be processed and how the results might be presented to the user.

Queries 1 & 2 are examples of MAP queries (Section 2.1.1). The system is given a set of observations (*e.g.*, gender, age, location of tumor) and asked to determine the most likely outcome for a target variable (*i.e.*, KPS and TTP, respectively). Observations can be specified either automatically using information extraction techniques de-

scribed in Section 3.2.1.3 or by having the user manually specify observations through the user interface, as discussed in Section 3.3. Upon executing the query, results can be visualized using a GUI component to display the most likely state. In addition, other variables in the model that influence the result may be identified by finding the paths of influence (Section 3.2.2.2) and highlighted.

The goal of Query 3 is to identify variables in the liver disorder model (Figure 4.3) supporting the outcome that the patient has primary biliary cirrhosis (PBC) given elevated bilirubin and GGTP levels. Answering this query involves examining the disease model structure to identify paths of influence (Section 3.2.2.2) where information flows between the observed variables (Total bilirubin, GGTP) and the target variable (PBC). The paths of influence are used to identify a subset of variables in the model that determine which data elements are displayed. In addition, strength of influence (Section 3.2.3.1) can be computed among the selected variables, which allow them to be ranked and highlighted based on how strongly they affect the target variable.

Query 4 asks the Bayesian belief network to identify a single document in the patient record that best summarizes the patient's treatment regimen (*i.e.*, medications, surgery) for glioblastoma multiforme. Semantic grouping and query expansion (Section 3.2.1) are used to answer this query: 1) a subset of the disease model is selected by identifying variables that belong to the semantic groups *Chemicals & Drugs, Devices,* and *Procedures*; 2) the selected variables are then mapped to matching concepts that have been identified in the patient record; and finally, 3) each document is ranked

based on a term frequency inverse document frequency (tf-idf) weighting scheme. The top ranked document is distinguished by increasing the transparency of all other documents.

Query 5 exemplifies how a user model (Section 3.4.2) can specify data elements that are relevant to a radiologist. Typically, radiologists are interested in viewing a patient's past imaging studies, associated radiology reports, and other clinical documents to aid during interpretation tasks. A set of filters is generated based on the user's profile; these filters constrain what types of information are displayed (*e.g.*, imaging studies, radiology reports). The query also specifies a temporal filter: only information that corresponds with the time period that the patient received erlotinib is displayed. The filter is translated into an instance of the "include based on recent activity" rule (Section 3.5.3.1), which is used to prevent data elements that occur outside of the specified time period from being rendered.

Query 6 illustrates how a graphical disease model can be used to perform case-based retrieval. The user-provided observations (location, lung cancer type, ECOG score) are used to instantiate the model; the posterior probability distribution for this instantiation is then compared to other cases in a database and assigned a similarity score (*e.g.*, using Kullback-Leibler divergence). Cases are ranked based on how similar they are to the inputted case. Case-based retrieval is further described as part of the VQI application in Section 4.2.2.3.

Query 7 demonstrates the utility of layering two types of information on a single display. The user first sets a temporal filter that removes all of the events that occur prior to the patient receiving dexamethasone. Then, leveraging the inclusion rules specified in the visual dictionary (Section 3.5.3.1), the "include based on value" rule is invoked to show only the laboratory values that are abnormal during this time period. The "compose by superimposing" rule (Section 3.5.3.3) is used to overlay the medications on top of the laboratory test results.

Query 8 showcases how a BBN can be used to identify relevant data elements in the patient record. First, a model is generated that represents the different eligibility criteria (*e.g.*, staging, performance status, medical history) as variables in the network. Then, variables in the model are mapped to data elements in the patient record. Any matching data elements are given priority on the screen. This query is explored further in Section 4.3.4.

3.2. Characterizing Graphical Disease Models

Graphical disease models contain a wealth of knowledge embedded in their properties. The following sections discuss approaches for obtaining contextual information from each aspect of the model. First, the process of mapping variables in the disease model to data elements in the patient record and to other knowledge sources is examined. Next, attributes that characterize the structure of the model (*i.e.*, node degree and path length, paths of influence, Markov blanket) are explored. Finally, two approaches (*i.e.*,

strength of influence, value of information) for quantitatively measuring the influence that variables have on each other are described.

3.2.1. Variables

The initial step in constructing a BBN is to identify those variables that can be observed and those intermediate and output variables that need to be inferred. Considerations include: can the variable be practically measured as part of routine clinical care; and what are the possible values that the variable can take on? Models should be comprehensive in that they draw conclusions from an array of evidence. Selected variables should be carefully defined using standardized representations (*e.g.*, UMLS, RadLex) whenever possible; this allows each variable to be mapped across different lexicons.

Variables are represented as nodes in the graph. Three types of nodes exist: 1) nodes that represent an outcome, hypothesis, or values of interest (target nodes); 2) nodes that provide the ability to input information to the model (evidence nodes); and 3) nodes that summarize the effect of a subset of parent nodes on a child node (intermediate nodes).

The goal of mapping is to identify all data elements in the patient record that relate to each variable in the model. The general process is depicted in Figure 3.1. Variable names are first mapped to standardized concepts represented in a medical lexicon. The lexicon is used to retrieve all descendants and synonyms of the variable name that are used to fully identify related variables in the patient record. For instance, if a variable represents the concept of *tumor size*, the process attempts to find all data elements that

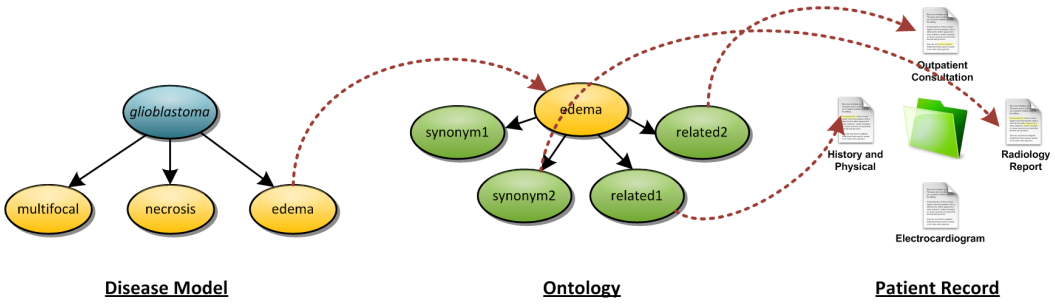


Figure 3.1: The process of mapping variables in the disease model to data elements in the patient record using an ontology.

explicitly or implicitly contain references to tumor size. Explicit references are those that directly mention the query term; for instance, a radiology report may contain explicit references to tumor size. Implicit references are those that indirectly refer to the term; tumor size may also be expressed by annotating an image with a measurement of the longest diameter of a lesion. The mapping process occurs in three steps: 1) variable names are first normalized; 2) the normalized names are then mapped to a medical lexicon where an expanded set of terms are generated through a process called *query expansion*; and 3) variable states are matched with corresponding attributes extracted from the patient record. The following sections describe each step in detail.

3.2.1.1. Mapping variables to concepts

The benefit of linking a disease model to the patient record is that it translates any manipulations done to the model into the identification of relevant data elements in the patient record. My approach utilizes the variable name; the name is typically descriptive of the concept that the variable represents (*e.g.*, disease name, finding, symptom, process, test). Intuitively, the process involves searching the entire patient record using

a variable name, and any matching instances found in the patient record are then mapped back to that variable. However, using a literal string search for the variable name to identify related terms in the patient record does not always result in sufficient or appropriate matches; a single concept can be expressed in multiple ways. In order to improve recall, a medical lexicon (*e.g.*, Unified Medical Language System, UMLS) is used to identify all of the variable name's synonymous and related terms. This expanded set of terms is then used to search the patient record, and matches are mapped to that variable using a process called *query expansion*.

Query expansion. Query expansion is the process of adding related search terms to the original query with the goal of improving the ability to recall of related, specialized terms that may not have been exact matches to the original query. Query expansion has been widely researched and applied [93]; in medicine, it has been used to improve the retrieval of biomedical documents [94, 95] and to aid consumers with searching for health information [96]. Different types of query expansion exist: term expansion and concept expansion. Term expansion is used to extend a strict literal search of a word or phrase by including word variants for plurals, possessives, hyphenation, compound words, and alternative spellings. In concept expansion, a medical lexicon is used to expand the query term to include descendents of the original term and synonyms. [97] illustrates one application that uses concept expansion for data retrieval: using a proprietary knowledge source called MED, the authors demonstrate how the original query term *heart* is expanded to include related disease terms (*heart*

diseases), laboratory tests (*cardiac enzyme test*), and radiology reports (*chest x-rays*). This expanded set of terms is used to query the patient record to identify any patient data that is relevant to *heart*. [95] demonstrates an approach for query expansion that utilizes the UMLS Metathesaurus to provide a list of related terms and the Semantic Network to define the relationships between these terms. The authors show that retrieval performance can be improved by expanding the original query using the term's direct parents, children, and siblings as defined in UMLS. While my approach follows that of [95], it differs in two ways: 1) the focus of my work is on using the graphical disease model to generate the original query; and 2) query expansion is used as a tool to improve the performance of mapping variables to relevant data elements. The process of query expansion is summarized as follows:

1. Variable names are first normalized. Term normalization converts singular and plural nouns to a standard form. It also converts words to a preferred spelling, stripped hyphens, stripped possessives, and the shortest form of compound words. All letters are lowercased and white spaces are converted to single blanks.

2. Then, variants (*e.g.*, plurals, possessives) of the normalized term that may exist in the patient record are found.

3. The normalized term is mapped to a matching concept in a medical lexicon. Querying the lexicon provides information about synonymous terms, parent (broader) terms, child (more specific), and sibling (similar) terms.

4. The entire patient record is searched using this expanded set of terms. Text data are parsed to find any matches with the expanded set of terms. Header information stored as part of medical images can also be parsed to identify relevant images. Metadata associated with laboratory values can be used to identify relevant test results. For all terms that are found in the record, the location of the term is retained in a data structure associated with that variable.

While query expansion is not a new concept, I have applied it towards linking data elements in the patient record to variables in the disease model. This is an important initial step because once variables and documents are linked, manipulations to the disease model can be used to identify relevant subsets of data in the patient record. The precision and recall of using query expansion to map variables to documents in the patient record was measured in the context of retrieving relevant documents. The goal of this study was to determine whether UMLS was comprehensive enough to find all of the variable names represented in the model; results are discussed in Section 5.2.2.

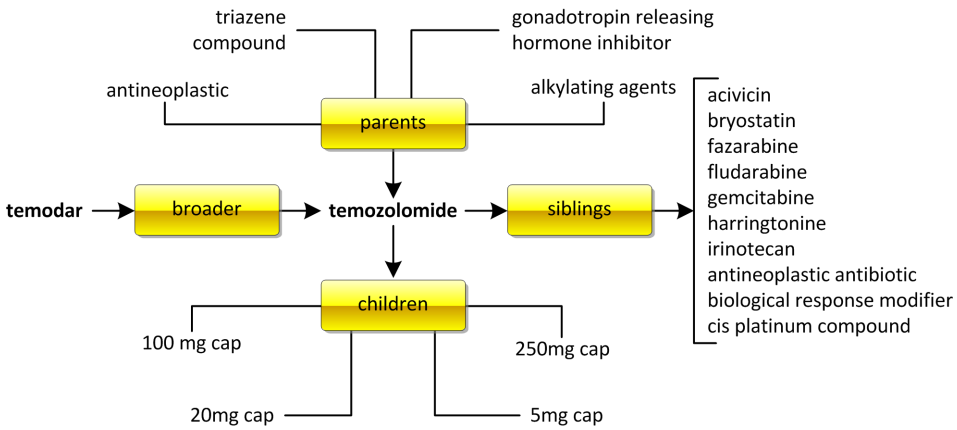


Figure 3.2: The result of concept expansion performed for the term *Temodar*.

Consider Query 3 from Section 3.1: to execute this query, a list of treatment-related terms is needed to aid in the identification of documents that discuss treatment history. First, query expansion is used to generate related terms to treatment-related variables in the disease model. If we examine the term *Temodar* as an example, UMLS can be leveraged to perform query expansion; the results are illustrated in Figure 3.2. Query expansion identifies four classes of terms: broader terms (*i.e.*, temozolomide*)*, parents (*e.g.*, antineoplastic, alkylating agents), siblings (*e.g.*, acivicin, bryostatin), and children (*e.g.*, dosage amounts). This expanded set of terms is used to search and rank all of the clinical documents based on the number of matches. For Query 4, the document with the highest number of matches is returned to the user.

3.2.1.2. Grouping variables

The goal of grouping variables together is to identify subsets of variables in the model that have similar properties or meaning. Semantic groups facilitate tasks such as: 1) identifying all semantically-related variables and concepts (*e.g.*, find all variables pertaining to medications); and 2) finding all variables that pertain to a particular concept (*e.g.*, disease). As in the case in query expansion, a medical lexicon can be used to assign variables to a semantic group. UMLS includes the Semantic Network, which defines the relationships between variables. Another source, Medical Subject Headings (MeSH), is a controlled vocabulary that is used to index biomedical articles from MEDLINE. In MeSH, terms are organized hierarchically from most general to most specific in structures called "MeSH trees." The topmost level consists of fifteen topics,

each representing a unique area (*e.g.*, anatomy, organisms, diseases, chemical and drugs). Increasingly specific terms are then placed as child or leaf nodes in the tree. Given its generalization-specialization organization, MeSH trees can be used to group related variables in the model. Variable names can be first mapped to corresponding MeSH terms; then, the MeSH tree is used to identify any common parents that terms share.

In context-sensitive visualization, semantic grouping is used to identify subsets of relevant variables that can be visualized together. Based on context, groups of semantically related data may be placed in the same area of the screen or visually differentiated from other data elements by using the same background color. For example, in VQI (Section 4.2), semantic groups are used to help a user query a disease model by suggesting semantically related variables based on the user's current selection. If the user is formulating a query and is interested in inputting information about the patient's age, the system can use the semantic group to suggest other variables (*e.g.*, gender, ethnicity) that may be of interest to the user's query. In addition, parts of the patient record that are semantically related can be displayed in the same area of the screen. A rule can be defined to specify that any data element associated with the semantic group of demographics is always included in the EMR display and rendered at the top of the screen.

3.2.1.3. Assigning variable states

The patient record provides information that can be used to instantiate variables in the model. First, the relations between terms and attributes need to be defined and extracted from the patient record. Information extraction techniques such as natural language processing (NLP) may be used for this task. For example, the frame-based representation that is outputted by a NLP system provides a means for relating concepts and attributes. A frame-based output is an object-oriented representation of the information extracted from the document. If the user has a laboratory result "serum glucose test, 32mg/dl", the frame representation for this statement would be the concept *serum glucose test* with several attributes: 1) it is a lab test; 2) it measures glucose; 3) the specimen is a serum; 4) the units are "mg/dl"; and 5) the value is 32. The second step is to use the values in the frame representation to set the corresponding variable to a specific state. For example, information for instantiating tumor size can be found in radiology reports, which typically document the length of the longest diameter for tumors seen in imaging studies. However, sizes are typically recorded as a value on a continuous scale while the states in the disease model are categorical (*e.g.*, small, medium, large). Therefore, instructions for mapping values to states are needed. My solution is to store a mapping attribute as an annotation in the disease model. The attribute specifies the ranges of continuous values that make up a particular state. For example, in tumor size, the "small" state could be instantiated if the value of the tumor size as reported in the clinical document is less than 1 cm. The mapping is generated at the time the continuous variables are discretized during the model creation process.

3.2.2. Structure

The network topology of a Bayesian belief network captures the relationships that exist among variables in the model. The structure is comprised of a set of edges that connect variables together. These edges can be conceptually interpreted as linking causes to effects; for instance, an edge between age and time to survival can be interpreted as *age has an effect on time to survival*. The way variables are structured in a disease model often reflects the prevailing belief of domain experts or conclusions made in scientific literature. The network structure of a graphical model is a combination of probability and graph theories. From a purely graph theory standpoint, analysis can be done to examine how nodes are connected to one another; such work has already been explored in many areas including sociology, bioinformatics, and knowledge engineering [98]. Structural analysis can be performed on models to determine node degree, path length, and clustering coefficients. In addition, because graphical models encode probabilistic dependencies between nodes, the flow of information between nodes can be quantified by examining the models' associated probabilities; this leads to the useful property of conditional independence. Using the available probabilities, analysis of the model can be done to determine which variables in the model become independent of (and irrelevant to) the target variable given a set of observations.

In this work, I examine three approaches for quantifying how variables are related to one another through characterization of the model structure: the first approach

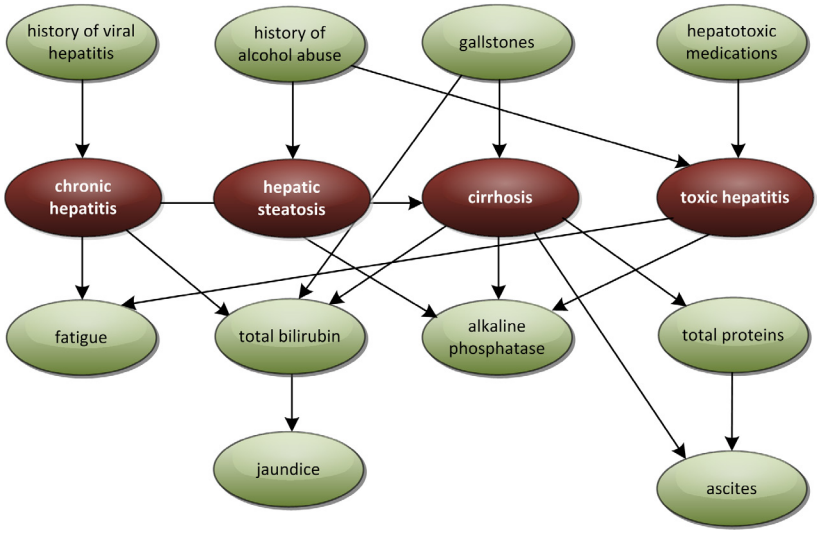


Figure 3.3: A subset of the multiple-disorder diagnosis version of HEPAR II [2]. Shaded (dark red) nodes represent output (disease) variables.

examines how a node is connected to all other nodes in the model. The second approach utilizes the notion of conditional independence and d-separation to identify when certain groups of variables are independent of other variables in the model. Finally, the third approach is a specialization of the second approach where a subset of variables is identified that provides all of the information to fully predict the behavior of a target variable. I illustrate these approaches in the context of Query 3 discussed in Section 3.1. A relevant subset of the liver disease model is presented in Figure 3.3.

3.2.2.1. Node degree and path length

Each node can be characterized by the connection it has with all of the other variables in the model. Here, two structurally-derived properties of the network are examined: node degree and path length.

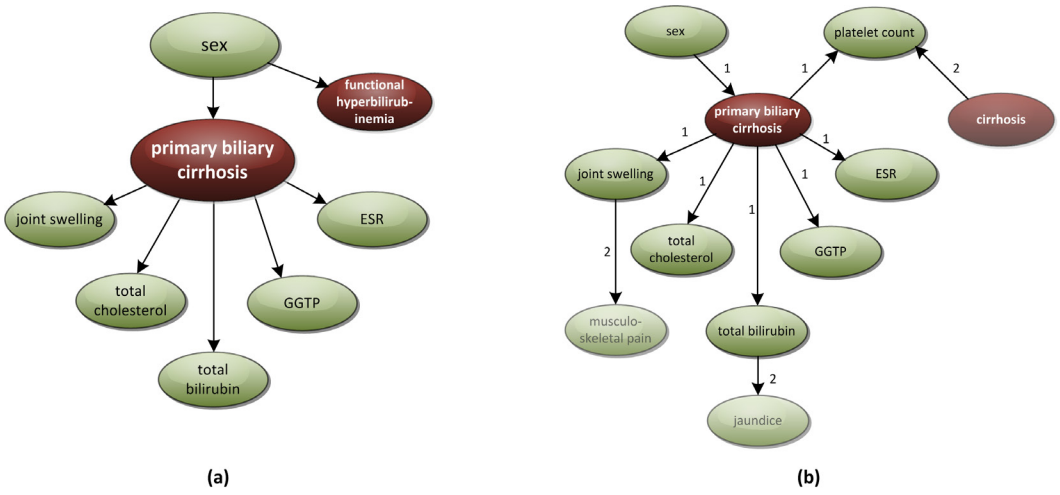


Figure 3.4: (a) A model representing the influences and effects of primary biliary cirrhosis. The size of each node is proportional to its in- and out- degrees. **(b)** Path lengths for nodes in the model based on their distance from primary biliary cirrhosis. More distant nodes from PBC are rendered with increasing levels of transparency.

One basic measure is to identify variables that are clustered together through a high degree of connectivity and variables that are dependent on a large number of parents, or conversely, serve as a parent to a large number of other dependent variables. These relationships can be identified by computing the *in-degree*, which is the number of incoming edges, and *out-degree*, the number of outgoing edges for each variable. Calculating the in- and out- degrees is done by counting the number of parents, children, and their combination for the node's total connectivity. The node degree is used to change the size of the corresponding data element: the amount of screen space that the visual representation uses is proportional to the variable's node degree (Figure 3.4a). In addition, filters can be used to specify whether to change the variable size based on the number of parents or the number of children.

The second measure for characterizing the relatedness of variables in a model is to compute the path length between the target variable and all other variables. Similar to distance analysis in social networks, the distance between two variables in a graphical model is measured by the minimum number of hops required to go from the origin node to the destination node. For example, examining the model depicted in Figure 3.4b, the distance between primary biliary cirrhosis (PBC) and total bilirubin is one hop, but the distance between PBC and jaundice is two hops. Using the target variable as a starting point, all other variables in the model are assigned a quantitative value that measures their distance from that variable. In addition, thresholds may be defined to partition variables based on their path lengths. Scores above a certain threshold are categorized as *irrelevant* while scores below the threshold are categorized as *relevant*. The threshold is user definable, providing the user control over how much information is presented on the screen at one time. The path length for any given node can be used to filter out data elements. For instance, any variables that have a distance greater than a specified threshold (*e.g.*, two hops) are rendered with increasing levels of transparency (*e.g.*, for every hop, the opacity is reduced by 25% until the element is rendered as fully transparent). Path length is used to determine which GUI components are presented to a user in an application described in Section 4.2.

To compute the shortest path between two nodes, Djiskstra's algorithm [99] is used. Djikstra's algorithm is a graph search algorithm that produces the shortest path tree between two nodes. To provide context of how Djikstra's algorithm is used in this

work, let us refer back to Query 3. The query asks to find what information is needed to conclude that a patient has PBC. Djikstra's algorithm can be used to rank every variable in the liver disorder model based on how distant they are to the target variable, PBC. The algorithm is summarized below:

```
CALCULATE_PATH_LENGTH(Graph g, Node source)
    for each node v in g
      dist[v] := inf
      previous[v] := null
    dist[source] := 0
    X := all nodes in g
    while X is not empty
      u := node in X with smallest dist[]
      remove u from X
      for each neighbor v of u
        alt := dist[u] + dist_between(u,v)
        if alt < dist[v]
          dist[v] alt
          previous[v] := u
    return previous[]
```

1. Prior to running the algorithm, every edge between two nodes is assigned a distance value. While the default value is 1, this value can also be based on the inverse value of the strength of influence defined in Section 3.2.3.1.

2. The initial node is set to zero and all other nodes are set to infinity.

3. The user specifies the target (source) node; this node is used as the starting point for the analysis and is set as the current node. For Query 3, the source node is PBC.

4. For the current node, consider all of its unvisited neighbors and calculate the distance from the initial node. If the distance is less than the previously record-

ed distance, overwrite the distance. From PBC, its immediate neighbors (*e.g.*, total bilirubin, platelet count, joint swelling) are visited; they all are assigned a value of one. The algorithm then iterates through the immediate neighbors (*e.g.*, total bilirubin) and visits their neighbors (*e.g.*, jaundice). For every hop that the algorithm takes, it increases the distance value. If the algorithm encounters a shorter distance between two nodes, the original distance is overwritten.

5. When done considering all neighbors of the current node, mark it as visited. A visited node will not be checked again; its distance recorded is final and minimal.

6. Set the unvisited node with the smallest distance from the initial node as the next current node and continue from step 4.

The result of Djikastra's algorithm is an array of distance values that represents the minimum distance (*e.g.*, number of hops) that is required to go to every node in the model from the target node; this is illustrated in Figure 3.4b. Therefore, a results display for Query 3 would present results for ESR, total bilirubin, joint swelling, sex, GGTP, and platelet count most prominently because they are each a single hop from PBC. However, data that pertains to musculoskeletal pain, cirrhosis, or jaundice are displayed with lower priority (and increased transparency) because they are more distant from PBC given the model's structure.

3.2.2.2. Paths of influence

Network structure can also be used to determine whether any independencies exist between two nodes. Independence is an important notion; in a Bayesian network, every variable is conditionally independent of its non-descendents given its parents (Section 2.1.1). From a computation standpoint, conditional independence assumptions enable the factorization of the full joint probability distribution, reducing the number of parameters that are needed to be assessed when performing inference. Qualitatively, conditional independence assertions can be used to determine what set of variables has influence on a target variable given some evidence. For the model depicted in Figure 3.3, if we are given information that the patient's total bilirubin value is significantly elevated above normal, then observing that the patient has jaundice is redundant; the jaundice variable becomes irrelevant because according to the model structure, it does not provide any new information that is not already provided by knowing the state of the total bilirubin variable. Hence, as the user interacts with the model by inputting information, the set of relevant variables dynamically changes based on what the current set of evidence and target variables that the user has specified.

This work uses the properties of conditional independence to identify significant variables and edges that contribute to the flow of information between the evidence and target variables. When a user poses a query to the model, the evidence and target variables represent a small subset of the actual variables that are involved in the reasoning. All of the remaining variables that are not conditionally independent of these va-

riables influence the result. Therefore, the approach discussed in [11] is used; it leverages the properties of d-separation to identify the paths in a model that information is able to flow from the evidence variables to the target variable. The algorithms for identifying influential paths are summarized below:

```
FIND_INFLUENTIAL_PATHS(Graph g, Node[] evidence, Node target)
   for each <node e, state of e> in evidence
     N := all parent and children nodes of e
     for each n in N
       if n is a parent node of e
         direction := outgoing
         FIND_PATHS(g, n, target, direction, (e,n))
       else
         direction := incoming
         FIND_PATHS(g, n, target, direction, (e,n))

FIND_PATHS(Graph g, Node current, Node destination, direction, path)
   if current is equal to destination
     add path to list of paths
   else
     N := set of parent or children nodes of current that does not
          d-separate current from other nodes in g
     for each n in N
       if n is not in path
         add n to the end of path
         if n is a parent of current
           direction := outgoing
           FIND_PATHS(g, n, destination, direction, path)
         else
           direction := incoming
           FIND_PATHS(g, n, destination, direction, path)
   return list of paths
```

The Find_Influential_Paths algorithm takes the directed acyclic graph (DAG), a set of evidence nodes, and a target node as inputs. It first iterates through each evidence node and attempts to find any paths in the DAG from that node to the evidence node which are not d-separated using Find_Paths. The output of this algorithm is a list of paths that facilitate the flow of information between the observed and target variables and the variables that lie along these paths. The paths of influence can be used to de-

termine which variables are of particular importance to a given query. As inputting evidence makes some variables redundant based on conditional independence, computing the paths of influence reduces the number of variables that are considered relevant for a given query and hence, lessens the amount of redundant data displayed on the screen.

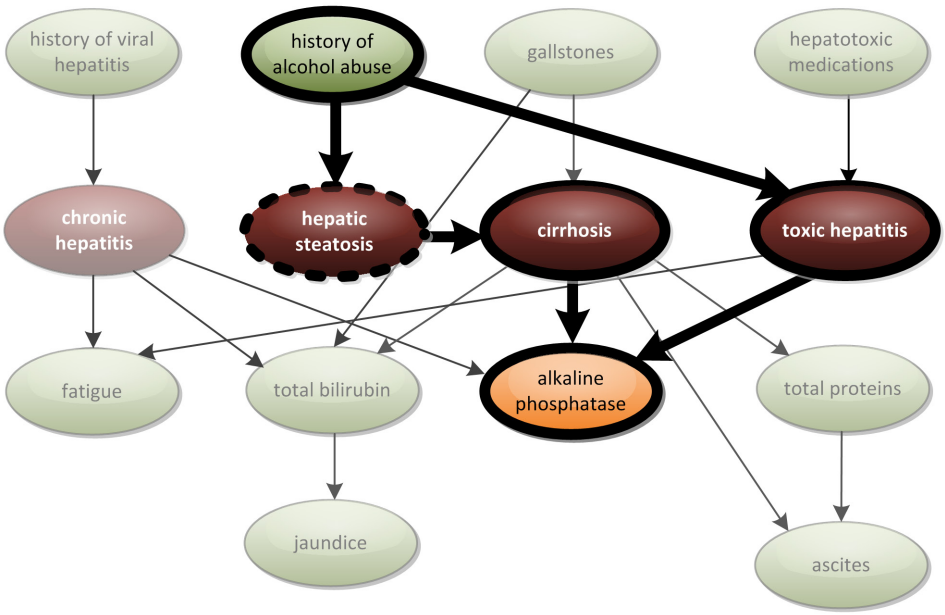


Figure 3.5: Example of identifying the path of influence for the target variable hepatic steatosis when given evidence about alkaline phosphatase. Outlined variables and edges represent variables that are part of the path of influence; all other variables have been made semi-transparent.

Let us consider an example using a subset of the liver disease model depicted in Figure 3.3. Assume that a physician wishes to determine whether a patient has a condition called hepatic steatosis. The physician has recently received the latest laboratory test results and notices that the patient's alkaline phosphatase is abnormally elevated. Using paths of influence, the system can identify other variables in the model whose val-

ues change given that the target variable is hepatic steatosis. The result of the algorithm is depicted in Figure 3.5: information can flow between alkaline phosphatase and hepatic steatosis through toxic hepatitis and history of alcohol abuse. Therefore, if information exists that is related to these variables in the patient record, it would be highlighted in the display.

3.2.2.3. Markov blanket

The Markov blanket provides a method for quickly identifying which variables in the model are necessary to be known before the target variable is fully characterized. It is used to select the initial set of variables that is presented in the GUI. Before the Markov blanket can be calculated, the user needs to select a target variable first; selection may be done by selecting a data element of interest that corresponds to a variable in the model. Once a target variable is selected, only variables in the Markov blanket are displayed on the interface; all other variables are made transparent. Referring back to the liver disease model, given that the user is interested in determining whether a patient has cirrhosis, an initial display can be generated using the target variable's Markov blanket, which consists of gallstones, hepatic steatosis, chronic hepatitis, toxic hepatitis, total bilirubin, alkaline phosphatase, total proteins, and ascites (Figure 3.6). Therefore, data elements that correspond to these variables are initially highlighted in the GUI. As the user interacts with the interface, then additional variables are identified using methods such as paths of influence.

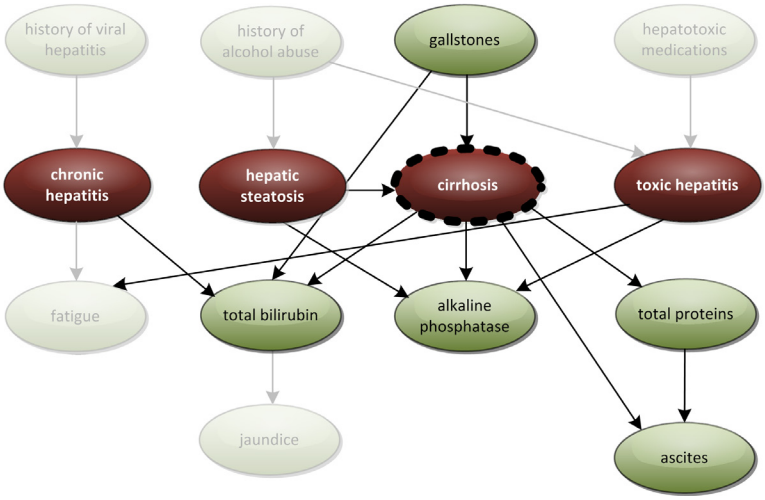


Figure 3.6: The Markov blanket for the variable cirrhosis. Variables outside of the Markov blanket are drawn semi-transparently.

3.2.3. Parameters

Each variable in the Bayesian network is associated with a conditional probability table (CPT). The CPT specifies the probability for each possible value of the selected variable given information about the variable's parents. These values are a reflection of beliefs expressed by a domain expert, published scientific literature, or data collected from a population of patients. In this work, values in the CPT are used to determine the strength of association between one variable and another. In conjunction with information from the network's structure, parameters specified in these tables provide insight into the relative importance of data based on the relationships and associated joint probabilities (*e.g.*, how does a change in X impact Y?).

Sensitivity analysis [100] is used to understand the impact that changing one variable has on the entire network. Such analysis permits inspection of how other evidence va-

riables are impacted. Sensitivity may be computed either analytically or by direct variation of a given variable's parameters. In the analytic approach, partial derivatives of the model output are computed with respect to each of the model parameters [101]. These partial derivatives, called sensitivity values, measure the sensitivity of model outputs to local changes in parameter values. In the direct approach, model parameters are changed within defined limits, and the recomputed model outputs are compared to original outputs. However, the number of tests that need to be performed given the combinations of evidence is prohibitive; in this work, the analysis is based on two alternative approaches: strength of influence and value of information.

3.2.3.1. Strength of influence

The strength of influence quantifies the amount of influence one node has on a neighboring node as specified by the presence of an edge. In this dissertation, a *dynamic* approach originally presented in [9] is used. A dynamic approach accounts for any observations that have been inputted into the model. A difference exists whether the parent or child node is observed: for example, if a physician determines that a patient has chickenpox, the patient more than likely exhibits skin rashes. If the physician only observes that the patient has skin rashes, a variety of causes may be hypothesized such as allergies, fungal infection, or eczema. Therefore, knowing that the patient has chickenpox greatly increases the probability of finding that the patient has rashes, but the probability of chickenpox given that the physician observes rashes would only increase slightly because multiple explanations are plausible. A static approach would

not have taken these differences into account. The strength of influence is computed by first calculating the posterior probability distribution of a node for each possible state of its parents and children nodes. Then, using the posterior distribution for each state, the distance between distributions is computed using the Kullback-Leibler (KL) divergence [102]. KL divergence measures the difference between two probability distributions *P(x)* and *Q(x)* where *x* is a specific state in a set of all possible states *X* using the following equation:

$$D_{KL}(P,Q) = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)}$$

The strength of influence value is computed by averaging all of the KL values that have been calculated for each state of a variable. For every new observation that is introduced into the model, the strength of influence for each variable is recalculated.

Strength of influence is used to quantify which pairs of variables are relevant for display. In this work, I explored ways to assign different colors and levels of opacity to a visual component based on how strongly it relates to a particular target variable. Given a user-defined threshold, any pair of variables that has a value below the specified threshold is made increasingly transparent. For the liver disorder model, if a user is interested in determining which variables have the strongest strength of influence on the target variable toxic hepatitis, the strength of influence analysis shows that the variable hepatotoxic medications has the strongest influence followed by alkaline phosphatase, and fatigue. If a user specifies a threshold of influence to be greater than

0.10, then for this example, the data elements that would be highlighted are related to hepatotoxic medications. All other variables are rendered with varying levels of transparency as depicted in Figure 3.7.

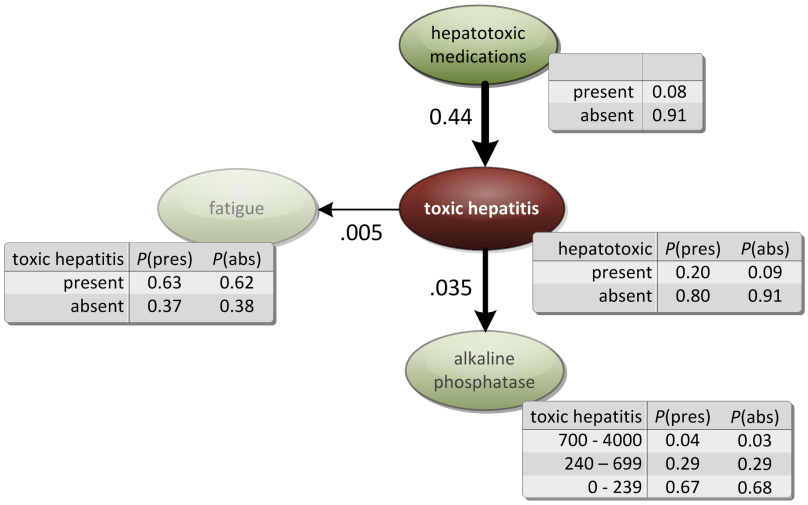


Figure 3.7: Strength of influence calculated for a small subset of the liver disorder model with toxic hepatitis being the target variable. Variables with a strength of influence less than 0.10 are drawn with increasing levels of transparency.

3.2.3.2. Value of information (VOI)

When performing diagnosis, a physician typically gathers as much relevant information as possible about the patient prior to making a final diagnosis regarding the patient's condition. Gathering such information is associated with some cost and benefit. The goal is to reduce the uncertainty regarding the value of some random variables in the user's decision model and thereby improve the quality of the user's action. For example, a physician may need to choose between obtaining a computed tomography (CT) scan versus a chest x-ray to screen patients for lung cancer; while CT scans are typically more accurate, they can cost much more than a chest x-ray. The question to

ask this model is: *would ordering a CT scan be worthwhile to check whether a patient with an extensive smoking history has lung cancer?* Intuitively, the value of any information source is defined as the difference between the utilities of two strategies: in the first strategy, the user is able to change the course of action based on information that is available and in the other, the user cannot make changes to a decision given the same information [12].

Utility-based VOI. Influence diagrams (Section 2.1.4.2) are useful for finding the set of decisions that optimize the outcome of an objective (*e.g.,* a patient's life expectancy). An influence diagram can be used to rank potential choices based on desirability using a utility function U(*a*) where *a* represents a decision. For instance, if a physician desires to select a treatment for a patient with GBM and needs to decide between ordering an MRI of the brain versus a biopsy, an influence diagram of this scenario can be used to determine the value of knowing the result of either the MRI scan or the biopsy. VOI calculations permit the determination of which choice (MRI scan or biopsy) provides a greater value of information for deciding what treatment (*i.e.*, radiation therapy or surgery) to choose. This calculation can be performed using the following utility function:

$$U(a \mid X = x) = \sum_{c} U(c) P(c \mid a, \mathbf{e}, X = x)$$

The variable *c* represents one configuration out of a collection of potential consequences and **e** represents the set of observed evidence. The utility of a single state of

the unobserved variable X is computed by selecting the configuration that provides the maximum result:

$$U(X = x) = \underset{a}{\operatorname{argmax}}\, U(a \mid X = x)$$

To compute the VOI of the variable, U_X, the result of U($X = x$) is averaged across each state in X and multiplied by P($X = x \mid \mathbf{e}$). Most value of information analyses are considered greedy, which means that the system determines the next best test by computing the value of information based on the assumption that the user will act immediately after seeing the results of the single test.

Non-Utility-based VOI. In many cases, a model for decisions and utilities may not exist. While BBNs can be translated into influence diagrams by assigning additional semantics to each variable and specifying a utility function, alternative formalisms can be used to assess VOI in traditional BBNs. Conceptually, the motivation for acquiring more information is to decrease the uncertainty of a hypothesis close to zero. Therefore, the goal is to assign more weight to probabilities close to zero or one when the uncertainty is low and assign less weight to probabilities in the middle area where uncertainty is high; these constraints are captured by an entropy function. Using the formalism defined in [103], if $P(T)$ is defined as the probability distribution for a target variable T, the equation to calculate the entropy of a distribution is:

$$Entropy(P(T)) = -\sum_{t \in T} P(t) \log_2 (P(t))$$

Then, the entropy-based value function $V(P(T))$ is:

$$V(P(T)) = -En(P(T)) = \sum_{t \in T} P(t) \log_2(P(t))$$

VOI analysis has been successfully applied to the area of user modeling: Lumiere [89] has used VOI analysis towards tailoring the functionality of a user interface to a specific user. The system consists of a Bayesian user model that predicts the types of functions that a user would need for a given task. VOI is used as a way to evaluate the costs and benefits of determining previously unobserved variables in the model that would be most valuable to evaluate. My work uses VOI analysis as a method for: 1) determining what data element would be most relevant for the given context, and 2) ordering which data elements are to be displayed first. VOI analysis can be used to answer queries such as: *given that a patient has a history of viral hepatitis, obesity, and an enlarged spleen, what would be the next best piece of information or test to obtain that would help rule out whether the patient has chronic hepatitis or cirrhosis?* The optimal course of action can be presented in the GUI as suggestions to the user regarding what the next course of action might be.

3.3. Query Specification

The usefulness of a belief network is its ability to answer a wide range of clinical questions using the knowledge encoded in the model. The process of computing the probabilities of each variable based on evidence is called *inference*. Inference com-

mences after the user instantiates the model by assigning one or more variables to a specific state. Common types of queries are reviewed in Section 2.1.1.

3.3.1. Evidence nodes

Part of the query process is specifying which nodes are observed; these nodes are then set to a specific state corresponding to the observed value. Depending on the types of information that the user selects, the observed nodes provide hints as to other related information that would be of interest to the user. For instance, if information is observed about the patient's symptoms and medical history, the system can identify which finding has the strongest effect on the outcome of the target node (*e.g.*, time to survival of patients with brain cancer). In this section, I examine how to compute the effect of evidence and how to determine whether an evidence variable agrees or disagrees with the result of a query. These methods are adapted from [11, 104] and used to rank the importance of data elements in the patient record on a specific task.

Mutual information measures how much knowing one of these variables reduces the uncertainty about the other. Consider a target variable, T, given evidence $E=\{E_1,\ldots,E_n\}$. The influence of a piece of evidence E_i on T can be measured in terms of whether and to what extent the change in distribution from $P(T)$ to $P(T|E_i)$ compares with the shift from $P(T)$ to $P(T|E)$. If the target variable has multiple states (*i.e.,* not a binary variable), then evidence variables that feed into the target variable can be classified into varying degrees of agreement: ones which are in agreement with the target variable, ones which are in disagreement, and ones that have mixed influence.

If we assign a target variable T and evidence variable E_i to states t_i and e_i, respectively, the information provided about $T = t_i$ by $E = e_i$ can be computed as $I(t_i;e_i) = \log(\frac{P(t_i \mid e_j)}{P(t_i)})$. If $I(t_i;e_i)$ results in a large positive value, it means that e_j strongly increases the probability of t_i; a large negative value means that e_j strongly decreases the probability of the target variable. For any state t_i in T, the probability shift produced by a single observation E_i can be classified as being either in agreement or disagreement with the shift produced by the set of evidence E by computing $I(t_i;E) \cdot I(t_i;E_i)$. The overall effect of a piece of evidence on the target variable is the sum of influence values for each state in the target variable:

$$\text{influence}(T;E;E_i) = \sum_{t_j \in T} I(t_j;E) I(t_j;E_i)$$

Again, let us examine Query 3 as an example. Assume that the physician does not believe the patient has cirrhosis and wants to check what available evidence supports or dispels his hypothesis. He determines that the patient has a significantly higher level of ESR reported, but total bilirubin levels are only slightly above normal and total cholesterol level is within normal range. The physician can compute the influence of evidence to determine what information supports his belief:

1. The liver disorder model is initalized without any variables observed.
2. Each evidence variable (ESR, total bilirubin, total cholesterol) is instantiated individually and the posterior probability for cirrhosis is calculated for each.

3. Using the posterior probabilities obtained, the influence for each evidence variable E_i is computed.

4. If the influence result for that evidence instantiation is greater than zero, then add E_i to the agree list; otherwise, add to the disagree list.

Each evidence variable is associated with an influence value, which can be compared against a threshold value to categorize variables into varying degrees of agreement: strongly agree, strongly disagree, or strongly mixed. In the past, this information has been visualized by varying the thickness of the edges in the DAG so that thicker edges represent a larger flow of information between variables. Also, changing the color of the edge has been explored such that nodes in agreement are assigned one color while conflicting nodes are assigned another. In this work, influence of evidence is used to identify how observed evidence extracted from the patient record relates to a given target variable. When the user specifies a target variable (either manually by user input or automatically by parsing a user model), influence of evidence is computed for each evidence variable in the model. The evidence is then rendered in a display with its size being proportional to how influential it is on the target. Different colors can also be used to denote whether the piece of evidence agrees or disagrees with the result of inference. In our example, both the total bilirubin and cholesterol test would be placed in the agree list, but the erythrocyte sedimentation rate (ESR) result would be placed in the disagree list. Data elements that disagree with the hypothesis could be visually differentiated by highlighted them in red.

3.3.2. Target nodes

Target nodes represent variables in the model whose most probable state is of particular interest to the user. These variables are typically ones that represent the outcome or diagnosis of a patient: examples of such variables in the domain of neuro-oncology include staging (diagnosis), time to survival (prognosis), and extent of resection (treatment). Given these semantics, the target node provides insight into the task that the user is trying to accomplish. For instance, if the user specifies lesion type as the target variable, it can be deduced that the user is interested in diagnosing the patient. Therefore, GUI components that are related to diagnosis can be automatically presented. Specifying a target variable is also important for characterizing the properties of the BBN; many of the algorithms discussed earlier calculate relevance and influence measures based on the relation between observed variables and a target variable.

Two ways exist for specifying a target node:

1. The user manually specifies the target node by selecting a data element and assigning it as the target node. The GUI can provide tools that allow the user to highlight a data element as a target node or display a dialog box that prompts the user to specify one.

2. The task model (*e.g.*, clinical guideline) can be defined to automatically specify which node is the target node for a particular task.

3.4. Other Sources of Context

Conceptually, context within healthcare can be thought of as being represented as a three-dimensional space with the axes defined by disease, user, and task [105]. A point in this space determines, for a given disease, user, and task, what set of constraints are used to customize the information that is displayed. Each point in this space specifies a unique instantiation of the composition rules (Section 3.5.3) that constrain how information is displayed. Attributes extracted from the graphical disease model provide information for only a single dimension of this space (*i.e.*, disease). In the following sections, I describe several other sources of knowledge are presented that can be used to either supplement or replace the graphical model.

3.4.1. Ontologies

Previous sections have detailed two roles that ontologies play in this work: 1) they identify related concepts through the process of query expansion (Section 3.2.1.1) and 2) they are used to classify variables into semantically related groups. In this section, I discuss a third role: ontologies provide domain knowledge in situations when the graphical model is not available. Graphical disease models are difficult to construct accurately and efficiently. Most models are frequently constructed by eliciting properties of the model from domain experts based on their experience and beliefs; creating these models are not only time consuming, but they also may be susceptible to the domain experts' biases. As a result, only a limited number of graphical disease models have been widely disseminated. It is in this context that I explore alternative sources

Relation	Attribute
Grade	
Is_Grade	Grade 4
Findings	
Excludes_Finding	Neuronal differentiation, precise histogenesis unknown
Has_Finding	Nuclear atypia, necrotic change, mitotic activity, microvascular proliferation, infultrative growth, anaplastic lesion
May_Have_Finding	Unfavorable clinical outcome, seizure, headache
Cell Type	
Has_Normal_Cell_Origin	Astrocyte
Has_Abnormal_Cel	Malignant cell, poorly differentiated neoplastic astrocyte
Anatomy	
Has_Primary_Anatomic_Site	Central nervous system, brain, nervous system
Molecular Abnormalities	
May_Have_Abnormal_Cell	Fibrillary neoplastic astrocyte, gemistocytic neoplastic astrocyte
May_Have_Cytogenetic_Abnormality	Loss of Chromosome 10p, del(10q2-26), gain of Chromosome 7q, del(10q23), loss of Chromosome 9p, TP53 gene inactivation, PDGFRalpha protein overexpression, PTEN gene inactivation, PDGFRalpha gene mutation, MDM2 gene amplification, EGFR protein overexpression, EGFR gene amplification

Table 3.1: Attributes and relations for the term brain glioblastoma (C4642) as specified in the NCI Thesaurus.

for domain knowledge, which can be used to characterize a disease and its associated attributes.

An ontology is a declarative model of a domain that defines any concepts, attributes, and relationships between concepts. I use the NCI Thesaurus (NCIT) as an example. As described in Section 2.2.1.2, the NCIT is a controlled vocabulary that is organized in multiple parent-child is_a hierarchies along with 100 distinct role relationships providing 135,000 asserted and inherited logical links between pairs of concepts. The ontology can be used as a disease model because like a graphical disease model, it captures the relationships between a disease and other related concepts at the molecular,

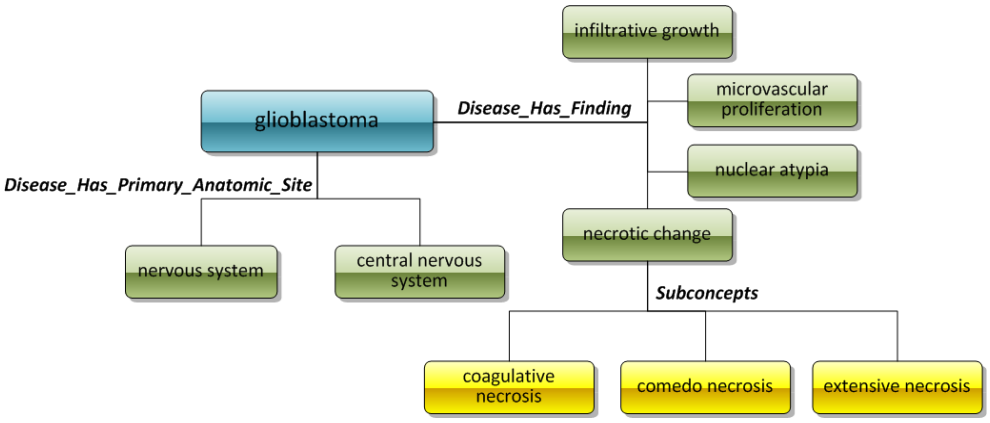


Figure 3.8: A subset of the NCI Thesaurus ontology depicting the concept glioblastoma and its associated terms and relations.

cellular, anatomic, morphological, and clinical levels. Consider the case of glioblastoma multiforme, which is represented as *brain glioblastoma* in NCIT. The concept is associated with multiple attributes and relations as listed in Table 3.1; two of the relationships, disease_has_primary_anatomic_site and disease_has_finding are graphically depicted in Figure 3.8. With an ontology, the same process of query expansion that is used to link variables in a graphical disease model to the patient record is now used to link concepts in the ontology. Ontologies and disease models can also be thought of as complementary: ontologies can supplement the knowledge in disease models by providing other important and related concepts that are not explicitly expressed in the graphical model. A major drawback to using the ontology is the lack of probabilistic values assigned to relationships between concepts. Therefore, ontologies cannot inherently quantify the strength of influence between two concepts; however, ongoing research in probabilistic ontologies may provide methods for overcoming this limitation

	PCP/Internist	Radiologist	Patient
Task description	Follow-up assessment	Image interpretation	Self-management
Clinical data			
Demographics	•	•	
Medical history	•	•	
Vitals (BP, BMI)	•		
AST, ALT, liver	•		•
Total cholesterol	•		•
HBV-5 panel	•		•
Imaging (abdominal)	•	•	
PCP reports	•	•	•
Hepatology reports	•	•	•
Radiology reports	•	•	
Medication history	•	•	•
Prioritization	1. Labs 2. Medical history 3. Medication history 4. Demographics 5. Vitals 6. PCP reports 7. Hepatology reports 8. Radiology reports 9. Imaging	1. Imaging 2. Radiology reports 3. Hepatology reports 4. Demographics 5. Medical history 6. Medication history 7. Labs	1. Labs 2. Medication history 3. PCP reports 4. Hepatology reports
Relationships	Medications → Labs Imaging → Radiology report	Imaging → Radiology report Medications → Labs	Medications → Labs
Visual metaphors	Labs ⇨Line plot Medical history ⇨List Medication history ⇨Time-line Demographics ⇨List Vitals ⇨Timeline Reports (all) ⇨Icon Imaging ⇨Icon	Imaging ⇨Presentation states Radiology reports ⇨Full text Hepatology reports ⇨Full text Demographics ⇨List Medical history ⇨List Medication history ⇨List Labs ⇨Table	Labs ⇨Line plot Medication history ⇨Time-line PCP reports ⇨Icon Hepatology reports ⇨Icon

Table 3.2: Example of a user model that is used to rank the types of data that would be of interest to each user group and the preferred visualizations that are used to present each data element.

[106]; implications of so-called probabilistic ontologies are discussed further in Section 6.2.

3.4.2. User models

Context-sensitive visualizations require an understanding of a user's preferences and information needs. This dissertation demonstrates how a user model can be applied in combination with a graphical disease model to enumerate the types of information and visualizations that are of interest to particular user groups.

My work employs a basic user model to capture the necessary information about a user's information needs: 1) what types of data sources are of interest to the user; 2) how should information be ranked on the screen; 3) which visual metaphors can be combined; and 4) how should visual metaphors be assigned to data elements. The model is patterned after [33] and is comprised of four sections; an example is provided in Table 3.2, which is used to illustrate each section:

1. Clinical data. This section of the model specifies the types of data from the patient record that are to be included or filtered out of the display. In the example, the user model captures the fact that primary care physicians require the most data: all of the data available about the patient's condition is displayed. On the other hand, for a radiologist, the model specifies that no laboratory data is to be displayed, only imaging data and clinical reports are to be shown.

2. Prioritization. Individual users have a preference on how they desire to view patient data. For certain user groups, the patient record should be dominated by the clinical reports; for others, the contents of the medical images determine what other data elements are viewed. Data elements in the user model are ranked based on these preferences. Higher ranked data elements (*e.g.*, laboratory values for primary care physicians, imaging studies for radiologists) are given a larger area on the screen in comparison to lower ranked data elements.

3. Relationships. This section in the model links different data types together based on predefined associations. For example, when interpreting an image, radiologists

find having past radiology reports with images helpful for identifying previous findings and tracking changes over time. Defining relationships between data elements instructs the application which elements in the patient record should be displayed together. These rules are encoded in a set of presentation rules described later in this chapter to link disparate data elements together.

4. Visual metaphors. This section specifies the mapping between data elements and visualizations. Each user has a unique requirement for viewing data: some users desire a quick overview of the available data while others desire a view showing all of the raw data. This section allows individual users are able to define the level of detail that is presented on the screen.

In addition, the graphical disease model itself can be inherently tailored to different user groups. Different models may exist for modeling a single disease; each model contains different set of variables that characterize the same disease. For example, a disease model that primarily incorporates imaging feature-derived variables to predict patient outcome would suit the information needs of radiologists. On the other hand, a model that represents various finding and treatment-related variables to predict progression and survival of a patient would provide information useful for oncologists. In Chapter 4, I present two disease models for brain tumors, an imaging-centric model and a prognostic model for glioblastoma multiforme that are tailored for use by radiologists and oncologists, respectively.

3.5. Visual Dictionary

Given that knowledge can be extracted from the graphical disease model (Section 3.2), the second objective of this dissertation is determining how to use this information to automatically filter patient data and generate a tailored display. The limitations of current medical data presentations are that: 1) they are hard-coded to display information a certain way; 2) they generally do not adapt to the changing context of why a user is viewing the data; and 3) they are not designed to easily accommodate new data types or user tasks. This work proposes to create a framework, called the *visual dictionary*, which links these disparate pieces of information to generate an adaptive display that identifies which variables are important and how they can be organized and presented. The visual dictionary addresses the aforementioned issues by allowing applications to adapt their GUIs based on changing contexts of use. Initially, the database is populated with values based on the default instantiation of the graphical disease model. As the user interacts with the model and poses a query, the visual dictionary is updated to reflect the changes in user preferences. After each query is executed, the application sends a new request to the dictionary to obtain updated instructions on how to render the data.

The visual dictionary performs three tasks: 1) it maps data elements to available visualizations based on context (*e.g.*, medical condition, user); 2) it changes the appearance of metaphors based on relevance; and 3) it incorporates multiple data elements into one display by following a set of composition rules that are defined by the disease

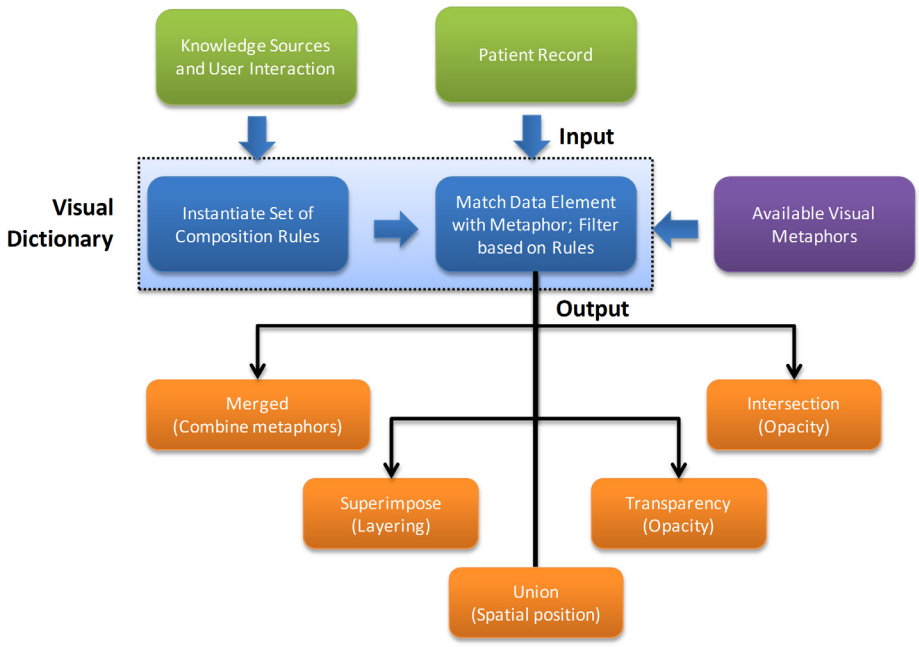


Figure 3.9: Flow chart depicting how the composition rules are used to generate instructions for presenting data elements in the clinical display. The inputs of the visual dictionary are user-specified filters to instantiate the composition rules and data elements in the patient record. The dictionary outputs a list of matched widgets and their appearance based on the specified filters.

model and filters. Composition rules provide instructions to an application for filtering and laying out data elements on the display based on a given context. The process of querying the visual dictionary is summarized as follows:

1. The entire patient record is pre-processed to identify and map relevant biomedical concepts and their attributes to a controlled vocabulary.

2. A user specifies a set of filters that define the context for viewing the data. These filters can be initialized based on the choice of medical problem, user group, and temporal window.

3. Given a specific context, the visual dictionary maps variables in the disease model to the concepts identified in the patient record using query expansion.

4. Each concept is associated with a visual metaphor. Its appearance is dependent on the properties of the disease model and dictated by the composition rules.

5. A summary display is generated by incorporating multiple visual metaphors together by following a set of composition rules that are influenced by the model and are in agreement with the parameters specified by the filters.

The process flow is illustrated in Figure 3.9.

3.5.1. Characterization of data elements

Data elements represent an atomic unit of meaningful information that exists in a patient record. They are comprised of an entity and value. An example of a data element is *gender*, which can be assigned the value "male" or "female." By themselves, data elements have little or no meaning. Knowing that the patient has been prescribed 10 mg p.o. b.i.d. of lisinopril is not clinically meaningful without additional information that helps place this fact in context with other elements in the patient record (*e.g.*, patient has a history of congestive heart failure). The process of data characterization helps provide context by linking individual data elements together that are semantically related.

In much of the real-world clinical environment, relevant data elements tend to be hidden in unstructured representations and spread across multiple different sources. Such

data requires preprocessing to extract relevant elements along with any associated attributes from the unstructured data. In this section, I review approaches for characterizing three sources of data: documents, images, and laboratory values. While many other types of data may exist in the patient record, a discussion about their implications is left to Section 6.2.

3.5.1.1. Documents

The first task is to identify which documents contain the most related information to a given concept (*e.g.*, medical problem) or criterion (*e.g.*, return all documented medications). This task can be accomplished by using the variables in a disease model as a way to identify relevant concepts to the disease that are in the document. As described in Section 3.2.1.1, variables in the model are mapped to terms in a clinical document through the process of query expansion. Once terms related to the variable are found in the document, a weighting scheme is used to rank how documents in a collection relate to a given concept or criterion. I use the *term frequency-inverse document frequency* (tf-idf) weighting scheme to measure how relevant a document is to a subset of variables in the model. This approach is widely used in information retrieval and has been applied towards clustering related documents in MEDLINE [107]. First, a subset of variables is identified from the graphical model. For instance, if the user is interested in finding documents that relate to brain tumors, the entire set of variables in the brain tumor model is used. In comparison, if the user is interested in documents that

relate to the medications prescribed for the patient's primary brain tumor, only a subset of variables that are assigned to the semantic group *Chemicals and Drugs* is used.

Calculating tf-idf proceeds as follows: suppose we have a clinical document and wish to determine which document is most relevant to the query terms *edema, necrosis,* and *mass effect*. The first step is to eliminate any documents that do not contain all three words. To further distinguish among the remaining documents, the number of times each term occurs in each document is counted and summed together; this value is the term frequency. The inverse document frequency is obtained by dividing the number of all documents by the number of documents containing those terms and taking the logarithm of the result. The inverse document frequency is used to diminish the weight of terms that occur very frequently in the collection and increases the weight of terms that occur rarely.

The equation to compute the tf-idf weight for a given set of concepts, w, is:

$$\text{tf-idf}_w = \frac{c(w,d)}{\max_v c(v,d)} \times \log \frac{N}{n_w}$$

where $c(w,d)$ is the number of times term w is found in document d. Depending on the number of terms that are found in a document, the tf-idf weight takes on a positive value, where zero means that no terms were found in the document. Using tf-idf weighting, subsets of variables from the graphical disease model are used to identify and rank relevant documents in the patient record using knowledge provided by the disease model or ontology.

3.5.1.2. Medical images

Medical imaging has become a predominant tool for objectively documenting patient presentation and clinical findings [108]. Patient care is largely dependent upon imaging to understand disease processes and to establish tangible evidence of response to treatment (*e.g.,* tumor getting smaller, aneurysm repaired, etc.). Thus, the effective integration of imaging data and related patient information is a necessity. Historically, the picture archiving and communication system (PACS) has limited users to query by certain keywords (*e.g.*, unique patient identifier, fields in the image header). However, these keywords often may not capture valuable visual characteristics contained within the image, thereby limiting the power of posed clinical queries and reducing the overall usefulness of the data. As imaging data becomes more abundant, methods are needed to extract and index the content of medical images so that they may be applied towards information retrieval and disease modeling. The ability to identify significant features embedded within the image data through automated extraction or manual delineation would provide a rich set of features that may be correlated with some outcome (*e.g.*, time to survival for cancer patients).

Similar to clinical documents, the purpose of medical image characterization is to extract relevant features that are used to index and link relevant studies to concepts in the disease model. To accomplish this, three types of information can be extracted from images:

- **Header information.** The Digital Imaging and Communications in Medicine (DICOM) standard provides for a common set of fields that are included with all data acquired by compliant imaging devices. The headers provide information related to each series in a study such as fields for description, modality, accession number, study date, anatomical part, and others. The same procedures to link variables with concepts in text documents are used to map variables to image attributes in a header file.

- **Image annotations.** DICOM Presentation State contains information on how a particular image should be displayed and additional annotations such as labels, window/level and zoom factor. This information is used to automatically identify slices of the image that are relevant to the user: images that have been annotated as being a *significant image* can be automatically rendered as a thumbnail and presented to the user. Other annotations such as longest diameter and volume measurements for tumors can be overlaid on the image to provide the user with context about how measurements in the radiology report are derived.

- **Image content.** This work would benefit from the advances in image understanding where methods are being developed to characterize images by color, texture, and shape. These features would allow the display to select properties such as tumor burden, tumor size, and other features that may be of interest to the user but not explicitly captured in the DICOM header.

The goal of my work in medical images is to map relevant features that have been extracted from the header, annotations, and image content to variables in the disease model. Then, the same analysis may be done to image content as what has been demonstrated with clinical documents to identify relevant subsets of imaging studies that are relevant for a given context. An application that uses the DICOM header to filter images is described further in Section 4.3.2.2.

3.5.1.3. Laboratory values

Clinical laboratory tests provide crucial data for a wide variety of medical decision making activities. Laboratory data is typically aggregated across laboratory and pathology departments and is used to assess chemistry, hematology, immunology, microbiology, genetic, and other histopathologic markers. Laboratory results are typically presented as numerical tables organized by name or date. Interpretation of laboratory data requires a comparison between temporal patterns in clusters of consecutive values within the same test and relationships between patterns across multiple tests. Result displays, as evidenced in Section 2.4.2, provide little assistance with either interpretation or decision-making.

While test results are an important piece of evidence in the medical record, when viewed alone, they do not provide the sufficient context to gauge how changes in result values affect the overall patient. Given other parts of the medical record, changes in test values may be associated with a particular clinical event (*e.g.*, the administration of a new drug).

Laboratory values are summarized in several ways:

- **Change in values**. A sudden increase or decrease in a laboratory value may signal underlying problems. Rate of change can be quantified by calculating the slope of change between values at two time points.

- **Abnormal values**. Results that are abnormal may provide insight into problems. Abnormal values can be categorized by degree. To identify abnormal values, the measured value of the test result is compared to the upper or bottom bounds of the normal range. The normal range is defined as values that are two standard deviations from the mean, covering values that have been attained by 95% of the normal population. Values that are over or under the normal bounds are considered abnormal; the degree of abnormality may be quantified as a percentage over or under the bounds of the normal value.

- **Recent values**. Physicians who are following up with a patient typically may not be interested in viewing the entire history of patient's laboratory values. Rather, they may only be interested in values that have been measured in the last month or since the last encounter. A temporal filter can be specified by the user such that any time points outside the specified window are not be rendered in the display.

These attributes directly correspond to the inclusion rules specified in the visual dictionary (Section 3.5). In this work, each test result is considered as an individual data element. Using hierarchical data clustering described in the next section, tests are

grouped into panels that have semantically related meaning: for example, low density lipoprotein (LDL), high density lipoprotein (HDL), and total cholesterol measurements belong to the lipid panel. Laboratory results are also presented depending on where they are mentioned in other parts of the patient record. For example, in outpatient consultations, the physician often summarizes the important laboratory results. These values can be hyperlinked to the actual test results.

3.5.1.4. Hierarchical data clustering

Data elements are clustered together using a common attribute through a process called hierarchical data clustering, which groups data (*e.g.*, contents of a patient's medical record) into increasingly specific categories. For instance, at the highest level of abstraction a patient record contains three types of data: documents, labs, and images. Further granularity is obtained by enumerating attributes within each type. For instance, clinical documents come in different forms (*e.g.*, letter, consult, pre-op/post-op), generated by different departments (*e.g.*, radiology, pathology), and written by various physicians (*e.g.*, radiologist, oncologist). Attributes are defined based on: 1) classification codes associated with a data element such as those published in ICD-9 (International Classification of Diseases) or SNOMED (Systematized Nomenclature of Medicine); 2) metadata encoding the constraints and characterizations of the data for a given medical entity (*e.g.*, biopsy and staging results from pathology); and 3) content-based features that are obtained through natural language processing and image understanding (*e.g.*, textual description of tumor border; size and texture of tumor

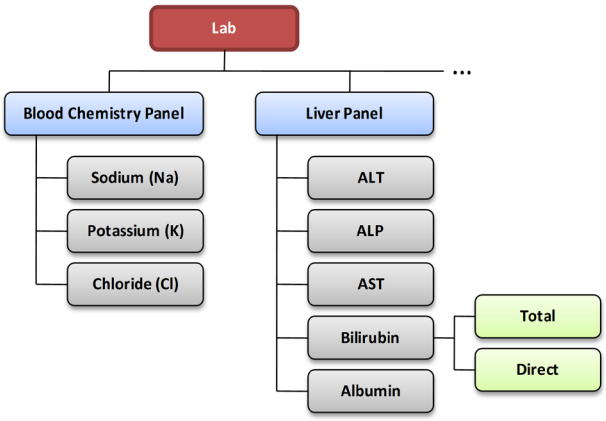


Figure 3.10: An illustration of the generalization-specialization relationship among attributes associated with laboratory results.

extracted from a computed tomography (CT) image). By hierarchically grouping data elements across multiple levels, the visual dictionary accommodates the depth and variety of information captured in clinical data. Data types have a generalization-specialization relationship, as illustrated in Figure 3.10. Under the lab data type, various panels are available (*e.g.*, chemistry, liver). Each panel is comprised of specific values (data elements) obtained from various tests (*e.g.*, amount of alanine aminotransferase (ALP) in the bile ducts), and each test may include several values (*e.g.*, bilirubin test results in two values: total and direct). The ability to organize data elements hierarchically is a key element in generating a tailored visualization. Higher-level abstractions are used to summarize the patient's data, while lower-level abstractions provide the details.

3.5.2. Mapping data elements to visual metaphors

For each selected data element, a visual metaphor is chosen with the goal of facilitating a user's understanding of the underlying trends and the importance of the informa-

tion in relation to the rest of the patient record. For instance, users who are presented with a large table of numerical values related to laboratory test results would be better served if they had a time series plot that allowed them to see the trends graphically and understand when values were increasing, decreasing, and abnormal.

Visual Metaphor	Types	Examples
Plots and charts	Line and scatter plot	EKG
	Bar chart	Histogram
	Radar chart	Comparing normal and abnormal clinical labs
Graphs and trees	Graph	Directed acyclic graph
	Flowchart	Medical guideline
	Tree	Ontology (*e.g.*, ICD-9)
	Dendrogram	Phylogenetic tree
Pictograms	Icons	Emoticons
	Maps	Anatomical atlas
	Diagrams	Velocity field for hemodynamic analysis
	Images	MRI

Table 3.3: A summary of the primary visual metaphors for presenting medical data.

3.5.2.1. Visual metaphors

As discussed in Section 2.3, medical data may be represented in different forms, depending on what type of information is to be conveyed and the context in which they are being shown. Table 3.3 summarizes these metaphors. Individual visual metaphors can span across multiple categories; for instance, tables and trees may be combined into a treetable, which organizes the rows in a table into a hierarchical tree. Because medical data is inherently temporal, the visualization of medical data can also be classified into one of two categories: point-based or interval-based. Point-based metaphors

are best used to summarize static information. They only allow the user to view the contents of an individual data element at a single point in time. When considering data collected at multiple time points, interval-based metaphors are necessary to summarize information visually across all time points. For instance, a time series plot is used to present trends for laboratory value results that are collected over the period of a year. For each selected data type, a decision is made to present either a point-based or interval-based metaphor. The selection is made based on information in the user model and whether the user has selected a temporal window (interval) or a specific time point.

3.5.2.2. Matching data elements to visual metaphors

The next step is to match data elements to appropriate visual metaphors. This section briefly describes the properties that are considered in this matching process.

User. The primary criterion that determines what visual metaphor to use is the user group. Depending on the user, different views on the same data may be required. In addition, users may desire different levels of detail, discussed below, depending on the type of data. A primary care physician may only be interested in knowing what imaging studies have been completed, so a thumbnail view of the images will suffice; however, a radiologist who needs to interpret these images requires a full image view with manipulation tools to aid them in reading the images.

Task. Each user group performs a unique set of tasks that have their own information requirements. A radiologist presenting a case to a tumor board would require different tools and set of information compared to one who is interpreting a patient's case and

dictating a report. The user model described in Section 3.4.2 provides a discussion of how the unique information requirements of each user is encoded in a knowledge base that is used to filter the types of information presented on the screen.

Data type. As discussed earlier, this work deals with three primary types of data elements: documents (textual), imaging (pictorial), and laboratory (numerical). The type of data places a constraint on the representations that may be used. For instance, imaging data will primarily be represented using a pictorial representation; laboratory data will likely be represented as a table, plot, chart, or graph. Knowing the data type restricts the types of associations that the visual dictionary can create between the data and available visual metaphors.

Dimensionality. The dimensionality of data reflects the number of variables that are included in a single visualization. Visualizations are typically limited in their ability to support multiple dimensions. Therefore, a method for determining whether a visual metaphor can support the number of variables to be visualized is necessary. For example, a list of patient medical problems represents a one-dimensional type of information; therefore, a list would suffice. On the other hand, for laboratory results that contain an array of dates and values, a two-dimensional representation such as a table or chart would be needed.

Level of detail. Depending on the user, data may be presented as either an overview or detailed view. An overview provides a summarization or abstraction of the information contained in a data element. For instance, rather than showing the user a full im-

aging study, which may consist of dozens of image slices, a thumbnail image of a representative slice maybe displayed. Another example would be laboratory results: rather than using a flowsheet to display laboratory values, the information may be summarized using a radar plot. In addition, level of detail specifies whether the point-based or interval-based representation is used. If the user is viewing static data, a point-based view is sufficient; but if the user is interested in finding trends in the data, an interval-based view would be necessary.

3.5.2.3. Common visual attributes

Visual metaphors are implemented as self-contained object-oriented components that render and provide methods for interacting with the data. While they contain all of the necessary instructions to execute and perform the necessary transformations from raw data to visual representation, all visual metaphors share a set of attributes that allow the user and application to modify their appearance and placement on the screen.

Opacity. The opacity of a metaphor can be changed based on how relevant the underlying data is to the user. For instance, data elements that are not at all related to the target variable may be made fully transparent and therefore not visible to the user. Other data elements that have varying levels of relevancy may be made proportionally less transparent therefore enhancing the user's ability to see the data. The goal is to make data elements that are highly relevant more visible to the user while other pieces of information are rendered in the background to some context but do not take the user's attention away from the primary evidence.

Layout. Each metaphor can be placed at a unique spatial coordinate on the screen. The position allows semantically related metaphors to be placed closer together while less relevant elements are further apart.

- Temporal layout. Data that has an associated timestamp may be laid out on a time-line display where the horizontal axis corresponds with the date and the vertical axis corresponds to different subclasses of information. A non-linear scale may be used to reduce any large gaps in time between data points (*e.g.*, if the patient has not seen the doctor in a long period of time).

- Graph layout. A multi-dimensional scaling algorithm called spring-embedding can be used to lay out widgets on the screen [109]. This algorithm accepts a matrix of weights that represent how closely or distantly related each data element is. Using this information, it attempts to create a set of points in a Euclidean space such that the distances between the points are proportional to how strongly the data elements influence each other. Each data element is connected by springs; the spring constant is set to be proportional to the strength of influence that one variable has on the other (*e.g.*, larger strength of influence values translate to higher spring constant). Attractive and repelling forces can also be specified by examining whether the evidence agrees or disagrees with a selected target outcome. The forces are then allowed to oscillate until it reaches a state with "minimum energy". The result of the algorithm is a set of points in space, where each point represents a data ele-

ment and the inter-point distances are a visual measure of relationships among the elements.

Layering. Visual metaphors may take advantage of the z-axis, allowing multiple metaphors to be overlaid on top of one another. For instance, laboratory values are typically uninformative when viewed by themselves, but placed in the context of the other data elements in the patient record (*e.g.*, clinical events, medications) the values take on additional meaning (*e.g.*, the abnormal rise in blood pressure is due to the administration of an anti-inflammatory drug). A combination of layering and opacity allows multiple metaphors to be combined, providing some reference for associations to be made between the two data elements. Layering is further addressed in the context of combining metaphors in Section 3.5.4.

3.5.3. Composition rules

Composition rules specify how a group of visual metaphors is presented to the user based on characteristics of the underlying data and graphical model. They define the boundaries that dictate when a data element is presented to the user and how its corresponding graphical representation is combined with others to generate a display. The purpose of composition rules is to constrain how visualizations are selected and combined in a display based on properties of the underlying data. Consider a scenario where a neuro-oncologist is interested in monitoring the progression of her patient's brain tumor. After the patient's medical record has been characterized, each extracted data element is then processed using the composition rules. Three types of composi-

tion rules are defined. *Inclusion rules* specify when a data element should be presented based on the values and trends in the data. If the patient recently had a blood test, those results would be presented, but results from a panel done two years ago would not. *Relevancy rules* use semantic information extracted from the model to determine what elements are considered relevant. Given that the oncologist is interested in the patient's brain tumor, only data elements pertaining to the tumor is displayed. Once the set of elements is defined, *presentation rules* generate a display by: 1) grouping similar data elements together based on similar characteristics; 2) layering and superimposing elements based on the relative importance of a data element compared to others; and 3) assigning a level of opacity.

3.5.3.1. Inclusion rules

Inclusion rules specify whether a particular data element is displayed given the characteristics of the data or contextual information about other variables being displayed. Bui et al. [81] proposed four initial rules:

- **Always include**, where values for a given data type are always presented. For instance, a patient's demographical profile (*e.g.*, age, gender, ethnicity) are standard pieces of information that are consistently displayed as part of the patient record; therefore, this data type is specified in the visual dictionary to be always included in the display.

- **Include based on recent activity**, where timestamps that are associated with data elements are used to determine whether a type of data is presented. Recent activity

is defined by the temporal filters that are specified by the user: the visual dictionary examines the timestamps associated with each data element and determines whether the data falls within the specified time range. For example, if the user desires to examine events related to the patient's GBM for the two months leading up to the first tumor progression, then only related data elements that fall within the specified time period are presented.

- **Include based on data value**, where the value of a given type of data determines whether it is presented. This rule is based on filters specified by the user to display any values that are above or below a certain threshold that correspond to a significant event. For instance, users can instruct the application to show only laboratory results with values that are above the normal range. Filtering data elements by value provides a tool for highlighting the results that are relevant for identifying abnormal events which may point to more serious underlying problems.

- **Include based on trend**, where changes in a series of values determine whether the type of data is presented. This rule is based on filters that compare the values of data elements over time and compare their rates of change. Physicians are commonly interested in noticing how values change over time; for example, a consistent rise in LDL and HDL values may indicate atherosclerosis. For changes that are characterized quantitatively (*e.g.*, tumor size, laboratory test value), a slope of change can be calculated; any sudden changes as denoted by a large slope may be brought to the attention of the user.

3.5.3.2. Relevancy rules

This work introduces additional relevancy rules that utilize the dependency knowledge extracted from the model structure to determine whether a data type is presented:

- **Include based on relation**, where variables that are closely related based on paths of influence or Markov blanket, are presented. If a physician wants to know whether tumor progression is the cause behind a measured increase in the patient's intracranial pressure based on relationships defined in the brain tumor example (Figure 2.5), any references in the patient record to mass effect, ventricular compression, and midline shift are identified and highlighted in the patient record.
- **Include based on significance**, where variables that are strongly influenced by each other based on the strength of influence or value of information, are presented. According to the liver disease model [2], hepatic fibrosis (excessive buildup of connective tissue in the liver) has a stronger strength of influence on whether the patient has cirrhosis compared to hepatic steatosis (excessive buildup of fat in the liver). Therefore, if the user is interested diagnosing patients with cirrhosis, the clinical display would emphasize references to hepatic fibrosis by using a combination of layering and highlighting.

3.5.3.3. Presentation rules

These rules, based on those originally proposed in [110], govern how multiple metaphors are combined into a single display:

- **Compose by merging**, where multiple data elements that have been determined to be relevant are displayed in the same presentation. For example, if the patient had values for potassium levels and calcium levels measured over time, these values should be plotted and rendered together in the same area of the display.

- **Compose by superimposing**, where multiple data elements are visually layered based on their strength of relation. Data elements corresponding to variables of stronger relevance are placed in the front while elements of less relevance are displayed in the back. If tumors have been segmented from images to track their changes in size, the contours could be superimposed on top of the image to assist users who are not experts at interpreting images with identifying the location of the tumor.

- **Compose by union**, where all data elements regardless of their values are rendered in the display. Composition by union may be performed on data elements that are semantically related; for example, data elements that correspond to the semantic type of *Finding* are grouped together by union and displayed using the same metaphor.

- **Compose by transparency**, where multiple data elements are rendered with an opacity that is proportional to their relevancy (as defined by the relevancy rules). For example, data elements that are more important to a certain disease based on the measures described earlier in this chapter have higher opacity than values that

are not as important; a more recent measurement may be displayed more opaquely than an older measurement.

- **Compose by intersection**, where the values of similar data elements are first compared with each other; only matching values are displayed. To illustrate, when matching the patient's symptoms with those common to a given disease, only the diseases that match the patient's symptoms are displayed while all other possible diseases are made transparent.

3.5.3.4. Application programming interface

Each presentation rule corresponds to a filter that can be selected by the user through the application programming interface. The following methods are defined for the visual dictionary:

- `Widget **setAlwaysInclude**(DataElement element, Boolean isEnabled)`
- `Widget **setIncludeRecent**(DataElement element, Date startDate, Date endDate)`
- `Widget **setIncludeValue**(DataElement element, Criteria criterion)`
- `Widget **setIncludeTrend**(DataElement element, Threshold threshold)`
- `Widget **setIncludeRelation**(DataElement element, Node[] source, Graph g, Integer threshold)`
- `Widget **setIncludeSignificance**(DataElement element, Node[] target, Graph g, Integer threshold)`

Once a data element has been assigned a visual metaphor (here, the implemented version is called a *widget*), then the presentation rules are used to combine and place widgets with respect to each other:

- `Widget merge(Widget firstWidget, Widget secondWidget)`
- `Widget superimpose(Widget topWidget, Boolean isTop)`
- `Layer union(Widget widget, Point position)`
- `Widget transparency(Widget widget, float alphaValue)`
- `Widget intersection(Widget widget, Criteria criterion)`

Common functions:

- `Array<Widget> sort(Array<Widget> list)`

3.5.4. Layers of the display

The display is comprised of multiple layers that are overlaid one another. Layering the display provides an additional dimension to convey information and relationships. Each layer is associated with a specific data type and ordered based on priority. Properties of the layered display are summarized in Figure 3.11.

Bottom layer. The bottom (or background) layer is the largest displayable area and is always fully opaque. It provides a common background and shared axis for all of the elements that are overlaid. For medical data, a typical background layer would be a timeline, providing a background for all other elements to be overlaid based on their

associated timestamp. However, depending on application, other types of backgrounds may be used. For instance, in VQI (Section 4.2), the bottommost layer consists of a normalized atlas of an anatomical region (*e.g.*, the brain); the atlas provides users with spatial context for formulating queries such as drawing and overlaying hypothetical tumors over the surrounding anatomical structures.

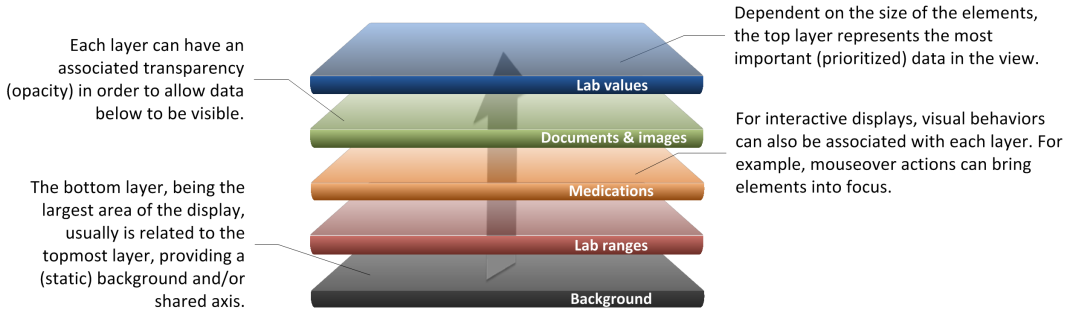


Figure 3.11: The display consists of multiple layers, each containing a specific type of data. Layers provide a means to use visual cues such as layering, opacity, and position to convey importance.

Middle layers. The middle layers may consist of different data types, depending on how data elements are prioritized for the display. Each layer is generated by combining metaphors together using the "compose by union" rule defined in the visual dictionary.

Top layer. The top layer contains metaphors that correspond to the data elements that have the most priority. These data elements are related to the variables that have the highest significance to a given context: for instance, if a physician is trying to determine the prognosis for a patient with lung cancer, the topmost layer would display metaphors that have the highest significance (as measured by strength of influence) in

terms of prognostic value (*e.g.*, staging, extent of resection, biological findings, metastases).

3.5.5. Integrated display

As touched upon in Section 1.2 and described in [33], the process of combining data elements into an integrated display in a context-sensitive manner can be summarized as the following steps: *identifying* the data that needs to go into the display; *relating* the data elements; *prioritizing* the selected data; *selecting* the appropriate visual metaphor for the data; and finally *laying out* the visual metaphors. Each stage is shapeable by context. To ground this discussion, consider the following data for patient with non-small cell lung cancer (NSCLC): 1) imaging studies including chest radiographs, CTs, and magnetic resonance (MR) images; 2) free-text and structured reports from physicians, including radiologists, thoracic oncologists, and the patient's primary care physician; 3) cytologic and histological analysis of a biopsy taken from the patient; and 4) the course of treatment during this period, including all medications and interventions.

Identifying data. In a medical display, the selection of data is dictated by the variables defined in a disease model that outlines the (clinical) information relevant to the condition's diagnosis and treatment. These models are typically all-encompassing; the scope of information needs to be honed by filters based on the inputs provided by a user and available knowledge sources. Different knowledge bases such as the graphical disease model can be used to steer the selection and filtering process. Similarly,

user models pinpoint elements of information needed to complete a task. In our example, identifying the data involves using knowledge about NSCLC that is provided by a graphical disease model or medical ontology to select data elements in the patient record that are related to variables or concepts in these models. For instance, if the model specifies that the drug gemcitabine is relevant to treating lung cancers, any reference to this drug in the patient's record is identified.

Prioritizing the data. Not all data elements are of equal importance in a diagnostic or treatment task. By ascertaining what is important, the amount of visual space allocated and prominence given to a graphical component can be gauged. The relative priorities do not necessarily correlate with the order in which data is accessed, but with where users' focus will linger most in the display. Context can help reveal clues as to what a given user or task may deem important. In my dissertation, a user model is employed to identify what concepts are relevant to the user and task. Revisiting Query 5 in Section 3.1, the query specifies that only information relevant to a radiologist should be displayed. Based on the user model in Table 3.2, data elements related to the imaging studies and radiology reports would be most relevant. Therefore, data elements are rearranged based on whether they are classified as either of these two data types; data that match these two types are displayed more prominently.

Relating the data. To facilitate physicians with the task of interpreting trends and patterns of interest within the patient's data, a display of clinical information should express a range of spatio-temporal and causal interactions. Being aware of the interplay

between data elements, an interface can better highlight potential relationships. Semantic relations available expressly (*e.g.,* rules encoded for a given medical problem) and indirectly (*e.g.,* relationships derived from the structure and parameters of the disease model) can be used to establish these linkages. From these connections, shared visual cues can then be used to demonstrate associations. Conventional cueing approaches include color coding of related data elements; spatial proximity (*e.g.,* overlapping, tooltips); similar line styles, etc. Continuing our example, a primary care physician viewing the lung cancer patient's data may be interested in seeing which medications (*e.g.*, erlotinib, gemcitabine) that the patient currently on and what affect they have on the patient (*e.g.*, by viewing the imaging studies). Based on the user model in Table 3.2, the user is presented relevant laboratory results alongside medications because adverse reactions to drugs can be detected by sudden changes in laboratory test results. In addition, imaging data is paired with their radiology reports to provide an interpretation of what is displayed in an image.

Selecting the appropriate visual metaphor. For each selected data element, a graphical representation is chosen that optimizes the viewer's ability to comprehend and interact with the data. Choosing the appropriate visual metaphor for a given set of data can be thought of as generating a sentence from a graphical language [111]: how do we best communicate the information to a given user? In Section 3.5.2.2, I described a set of attributes that can be used to determine which visual metaphor best represents a given data element. In addition, my work addresses how visual cues can be used to

highlight data elements in Section 3.5.2.3. For a radiologist interpreting a CT study to check whether new nodules are present in a lung cancer patient, the visual metaphor used to present the imaging study would provide all of the necessary tools to manipulate the images (window/level, zoom, annotate) and utilize DICOM Presentation States. On the other hand, when a patient views the imaging data, she is only presented with a simple overview (*e.g.*, detail-in-context view) of the data.

Laying out the data. The last step in integrating the data elements together is to spatially organize the information in the display. Here, the composition rules defined from the disease model and visual dictionary are helpful in guiding layout, with the intent of creating a visual/focal flow to the presentation and interaction with the data. The layout of the data takes into account possible visual interactions to exploit user search behaviors. The basic guideline of every layout follows Shneiderman's framework of overview, zoom/filter, and details on demand [112]: primary data in the display can be visible in a synopsis state, allowing for selection and augmentation with additional data (*e.g.,* but of lesser priority). Importantly, the ultimate source of information should accessible to the user. Layout not only comprises arrangement in the *x*-*y* plane, but also *layering* of the visual metaphors (*i.e., z*-order), allowing for juxtaposition of graphical elements. In the case of the NSCLC patient, visual metaphors can be collated and combined into a single layered view that facilitates a user's ability to see relationships. Data elements from different sources are overlaid using a combination of transparency and interaction to spatially relate information and support details-

on-demand behavior. For example, if a physician obtains results from the patient's histological study, then the results are overlaid with data from the patient's current therapy regimen to determine whether changes in chemotherapy is needed to better target the patient's specific grade of cancer.

CHAPTER 4

Applications

4. Overview

In the previous chapter, I described methods for characterizing graphical disease models to identify and select data elements from the patient record based on context. I also presented the visual dictionary as a data model for integrating information from these models with other knowledge sources to generate a set of composition rules that define how the graphical user interface (GUI) appears to a user. In this chapter, I describe how these concepts are implemented in two prototype applications. The first application, visual query interface (VQI), supports visual manipulation of the underlying graphical disease model and facilitates the user's understanding of the interplay between variables. The fundamental components for representing patient data and posing queries are called *graphical metaphors* [113], which correspond to interactive, visual representations of variables specified in the disease model. Interaction with the workstation is a closed-loop process: the underlying model influences what data elements are relevant and presented to the user. Continued user interaction with the graphical metaphors defines a new subset of variables in the model that guide the user with formulating queries. The second application, the adaptive electronic health record (AdaptEHR) viewer, is an integrated, longitudinal presentation of a patient record that dynamically filters data elements. The underlying knowledge sources are also used to

drive the adaption of the UI to specific users and tasks. I first ground the discussion with examples of graphical disease models in the domains of neuro-oncology and hepatology. Then, each application is introduced along with a brief review of related work. I discuss the primary components of each application followed by details of how the application is implemented. The results of user evaluations on these applications are described in Chapter 5.

4.1. Disease Model Examples

While the construction of disease models is not a focus of this dissertation, three disease models are described to provide a basis for understanding the application and evaluation of my work. The first model is a brain tumor prognostic model that incorporates variables derived from patient demographics, imaging data, and treatments. The outcome variables are time to progression (TTP), time to survival (TTS), and Karnofsky performance status (KPS). The second model captures the effect that magnetic resonance (MR) imaging features have on outcome variables. The final model is a probabilistic diagnostic model of liver disorders, called HEPAR II [2]. Variables are primarily obtained from medical history, physical examination, and laboratory results. The model diagnoses multiple liver-related diseases based on inputted evidence: hepatic steatosis, hepatic fibrosis, carcinoma, toxic/chronic/reactive hepatitis, cirrhosis, primary biliary cirrhosis (PBC), and hyperbilirubinemia.

4.1.1. Prognostic models of brain tumors

According to the National Cancer Institute, in the United States, over 21,810 patients are estimated to have been diagnosed with brain or nervous system-related tumors in 2008. Although there have been several decades of intense clinical and basic science research aimed at improving the survival of these individuals, life expectancy remains between 8-12 months. The practical reality regarding patients with primary brain tumors, particularly glioblastoma multiforme (GBM), is that managing this disease remains an enormous challenge given the tumor's aggressive nature. Researchers have begun to prospectively compile databases on brain tumor patients in the hope of improving both the understanding of the disease process and outcomes. The modeling effort utilized in this work draws upon data from the UCLA Neuro-oncology Clinic, which maintains a database of over 1,000 patients diagnosed with brain tumors. The database includes a variety of variables from sources such as the hospital information system (HIS), oncology and radiology reports, pathology results, imaging studies, and consultations.

Two models have been developed for predicting the time to survival of patients. In the first model (Figure 4.2 and Table 4.2), several types of brain tumors are modeled using features derived from MR images and from pathology to predict a patient's time to survival. In the second model (Figure 4.1 and Table 4.1), a subset of patients with GBM is used to capture the types of variables that a neuro-oncologist would examine and attempt to predict the progression and survival of a patient given this information.

Variables for both models are obtained from multiple sources: 1) oncology, radiology, pathology, and surgery reports; 2) expert opinion; and 3) reputable information sources (*e.g.*, Cochrane Collaboration, published literature). Variables can be grouped into five categories:

- **Demographics.** Past studies have shown that brain tumors occur more frequently in males than females and are slightly more common in Caucasians than other races [114]. These relationships have been modeled as the variables age, gender, and ethnicity.

- **Pathology.** The presence of certain pathological features such as multifocality and oligo component has been shown to be correlated with prognosis. Hence, these variables are represented in the model.

- **Imaging variables.** Imaging variables can be either quantitative (*e.g.,* tumor size) by characterizing features identified in the data or inferential based on descriptions provided by a domain expert (*e.g.,* as part of the radiology report for the clinical imaging study).

- **Treatments.** A select number of treatment variables have been modeled. Several popular chemotherapy treatments are represented: the primary drug is temozolomide (Temodar), which has its own node with states that represent common dosage amounts. Other chemotherapy drugs are represented in a single variable that represents the common combination of these drugs: PCV (procarbazine, lomustine

[CeeNu], vincristine), and carmustine. The model includes a variable that represents whether the patient has undergone radiation therapy including conformal and stereotactic radiotherapies.

- **Outcome variables.** The goal of the model is to predict the following outcome variables (shaded in dark red): TTP, KPS, and TTS.

The imaging-centric brain tumor model was parameterized using a set of 152 patients from the neuro-oncology database. These patients were chosen because they had the most complete collection of image features documented. To parameterize the GBM model, a separate subset of 200 patients from the neuro-oncology database was used. While some overlap exists between the two patient populations, the GBM model focuses exclusively on patients that have been diagnosed with GBM and received at least one dose of the chemotherapy drug temozolomide. The resulting models are illustrated in Figure 4.1 and Figure 4.2.

4.1.2. Diagnostic model of liver disorders

Cirrhosis is a chronic liver disease that is characterized by the replacement of liver tissue with fibrous scar tissues. This damaged tissue progressively leads to the loss of liver function. While alcohol is one of the primary causes of liver disease, other causes can include chronic hepatitis C, which causes inflammation and damage to the liver over time, hepatitis B and D, fatty liver disease, and autoimmune hepatitis.

HEPAR II [2] is a probabilistic diagnostic disease model for liver disorders. Common symptoms include the yellowing of the skin, liver enlargement, portal hypertension, buildup of fluid in the abdominal cavity, liver encephalopathy, and liver failure. Therefore, the diagnosis of liver disease typically requires a combination of data sources including the patient's history, results of a physical examination and laboratory tests, imaging studies using modalities such as computed tomography (CT) and MR, and a liver biopsy. From the collected information, 71 different variables were extracted and represented in the disease model. These variables can be categorized into five categories:

- **Patient history.** Several factors in the patient's medical history can influence whether a patient suffers from liver disease. These factors include past hospitalizations, surgeries, transfusions, injections, and alcoholism. In addition, other diseases such as diabetes can have an effect.

- **Physical exam.** As part of a routine exam, physicians record information such as whether the patient experiences upper abdominal pain, fatigue, or yellowing of the skin.

- **Laboratory tests.** Lab tests are important in gauging liver function. The liver panel includes tests that measure alanine aminotransferase (ALT), alkaline phosphatase (ALP), and total protein. Other tests such as the lipid panel and antibody tests (*e.g.*, Hepatitis A Virus Antibody IgM) are also helpful in narrowing a diagnosis.

- **Imaging variables.** MR imaging is particularly well-suited to evaluate liver pathology due to its ability to generate contrast (*e.g.*, gadolinium) to see fat perfusion or iron accumulation. From these images, features such as irregular liver edge and edema can be extracted and used to differentiate between different liver disorder diagnoses.

- **Outcome variables.** The goal of the model is to predict whether the patient has one of seven liver disorders: hepatic steatosis, hepatic fibrosis, carcinoma, toxic/chronic/reactive hepatitis, cirrhosis, primary biliary cirrhosis (PBC), or hyperbilirubinemia.

The complete liver disorder model is illustrated in Figure 4.3.

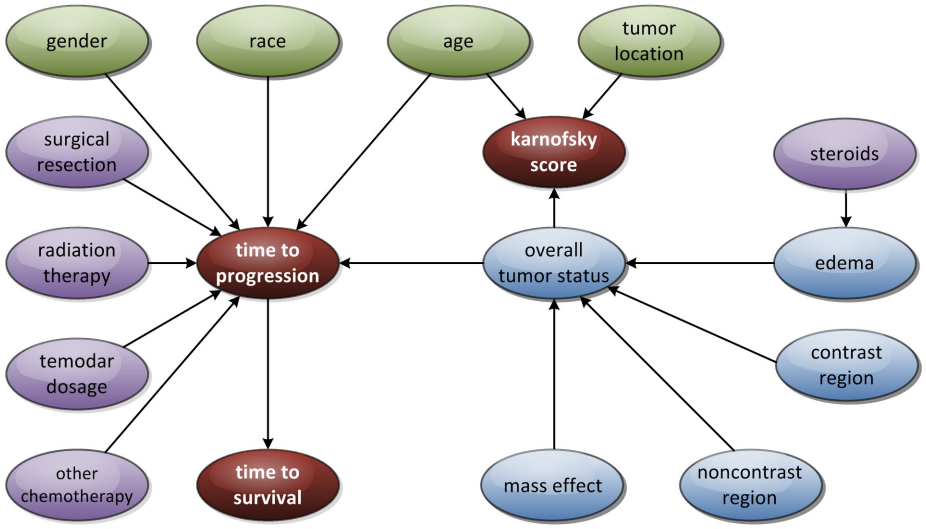


Figure 4.1: A Bayesian belief network for predicting the progression and survival of glioblastoma multiforme patients.

Evidence variable	Source	Evidence variable	Source
Gender (Male. Female)	HIS	**Time to progression** (< 2 mo, 3-6 mo, 6 mo-1 yr, 1-2 yr, >2 yr)	Oncology report
Race (White, Asian, Hispanic, Middle Eastern, Black, Other)	HIS	**Overall tumor status** (Worse, Marginally worse, No change, Marginally better, Better)	Oncology report
Age (17-50, >50)	HIS	**Temodar dosage** (6 states)	Oncology report
Edema (Worse, Marginally worse, No change, Marginally better, Better)	MR imaging study	**Other chemotherapy** (6 states)	Oncology report
Contrast region (Worse, Marginally worse, No change, Marginally better, Better)	MR imaging study	**Time to survival** (< 6 mo, 6 mo-1yr, 1-2 yr, > 2 yr)	Oncology report
Noncontrast region (Worse, Marginally worse, No change, Marginally better, Better)	MR imaging study	**Tumor location** (8 states)	Radiology report
Karnofsky performance status (<60, 60-69, 70-79, 80-89, 90-99, 100)	Oncology report	**Mass effect** (Worse, Marginally worse, No change, Marginally better, Better)	Radiology report
Steroids (6 states)	Oncology report	**Resection** (None, Biopsy, 20-90, 90-99, 100)	Surgical report
Radiation therapy (None, Regional, Stereotactic)	Oncology report		

Table 4.1: List of variables incorporated in the glioblastoma multiforme prognostic model. States (or number of states) for each variable are specified in parenthesis.

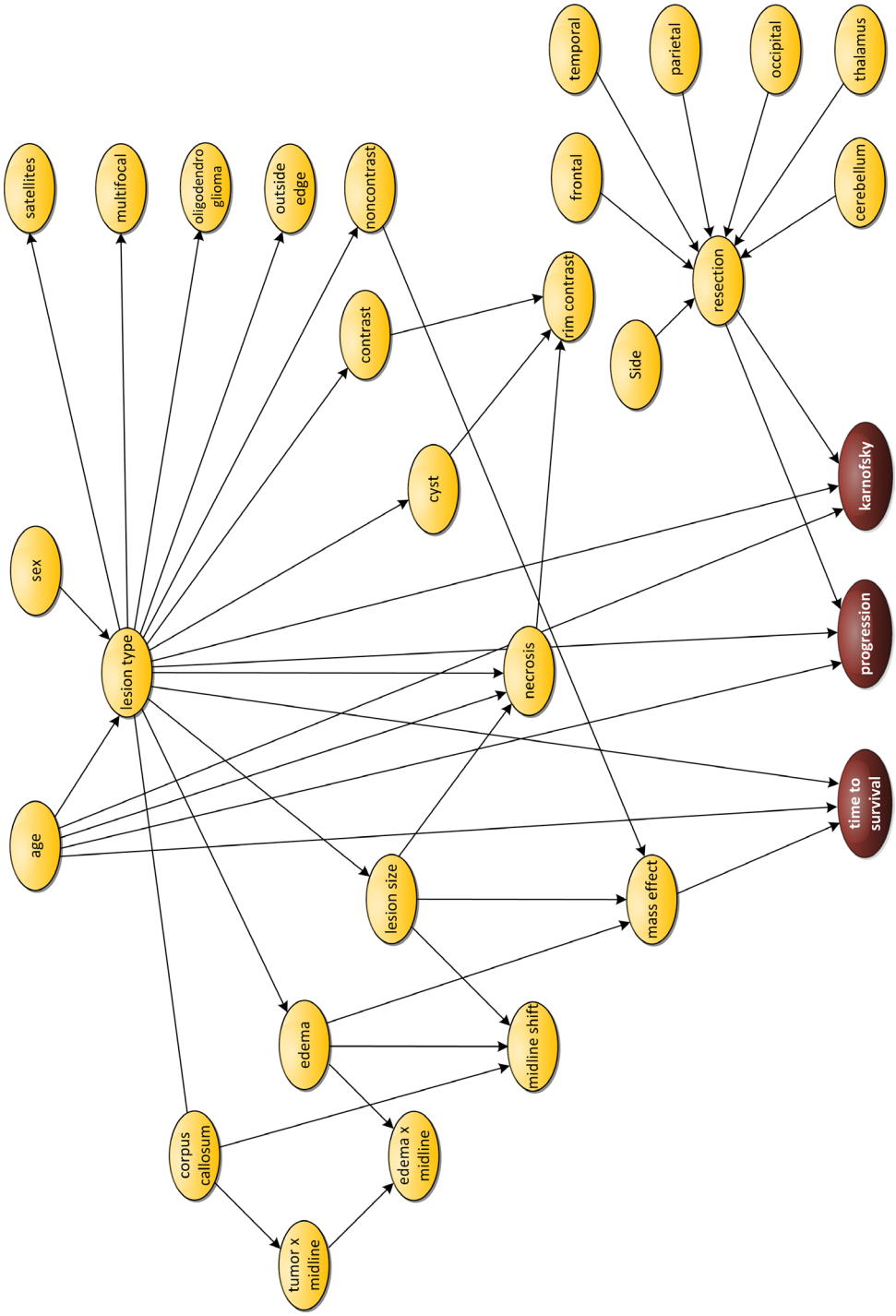


Figure 4.2: Bayesian belief network for predicting the progression and survival of brain tumor patients using variables derived from pathology and radiological features. Nodes shaded in red represent outcome variables.

Evidence variable	Source	Evidence variable	Source
Sex (Male, Female)	HIS	**Parietal Lobe** (Present, Absent)	MR imaging study
Age (17-50, >50)	HIS	**Necrosis** (Present, Absent)	MR imaging study
Outside Edge (Smooth, Irregular)	MR imaging study	**Lesion Size** (S, M, L, XL)	MR imaging study
Midline Shift (Present, Absent)	MR imaging study	**Noncontrast Enhancement** (Present, Absent)	MR imaging study
Thalamus (Present, Absent)	MR imaging study	**Cyst** (Present, Absent)	MR imaging study
Edema (None, Mild, Moderate/Severe)	MR imaging study	**Cerebellum** (Present, Absent)	MR imaging study
Rim Contrast (None, Thin, Thick)	MR imaging study	**Satellites** (Present, Absent)	MR imaging study
Tumor Extension X Midline (Yes, No)	MR imaging study	**Progression** (< 2 mo, 3-6 mo, 6 mo-1 yr, 1-2 yr, >2 yr)	Oncology report
Corpus Callosum (Present, Absent)	MR imaging study	**TTS** (< 6 mo, 6 mo-1yr, 1-2 yr, > 2 yr)	Oncology report
Side (Left, Right)	MR imaging study	**Karnofsky performance status** (< 60, 60-80,80-90,100)	Oncology report
Occipital Lobe (Present, Absent)	MR imaging study	**Oligodendroglioma** (None, Minor, Major)	Pathology report
Temporal Lobe (Present, Absent)	MR imaging study	**Lesion Type** (Anaplastic Astrocytoma, Anaplastic Glioma, Oligodendroglioma, Glioblastoma Multiforme)	Pathology report
Mass Effect (None, Mild, Moderate/Severe)	MR imaging study	**Multifocal** (Yes, No)	Pathology report
Contrast Enhancement (None, Partial, Solid)	MR imaging study	**Edema Extension X Midline** (Yes, No)	Radiology report
Frontal Lobe (Present, Absent)	MR imaging study	**Resection** (None, Biopsy, 20-90, 90-99, 100)	Surgical report

Table 4.2: List of variables incorporated in the imaging-centric model of brain tumors. States for each variable are specified in parenthesis.

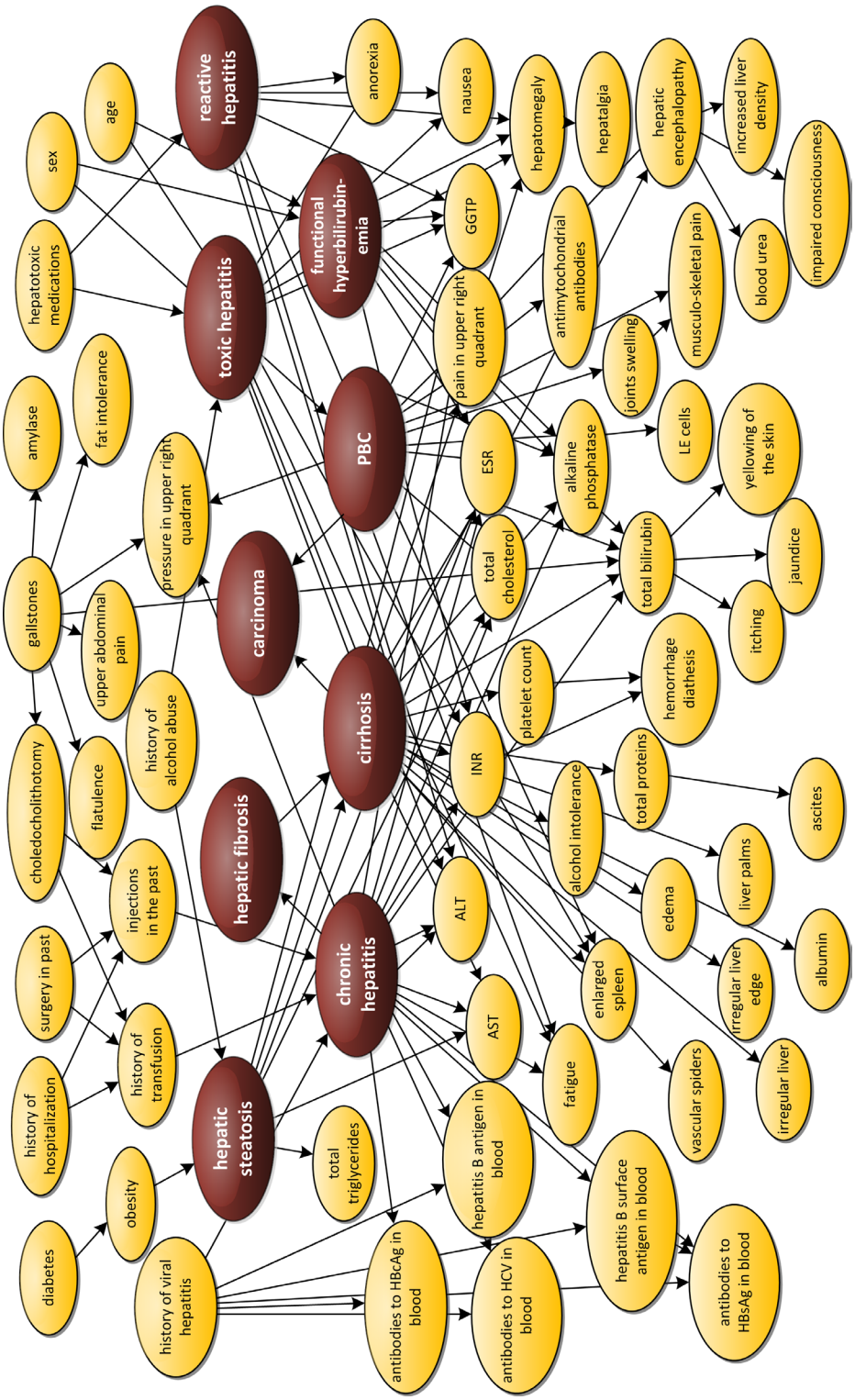


Figure 4.3: The variables and structure of HEPAR II [2]. Shaded nodes (dark red) represent outcome variables.

4.2. Visual Query Interface

The first application, called the *visual query interface* (VQI), facilitates inference on a disease model using a graphical paradigm. This system is designed for radiologists and other physicians who are interested in using image features (*e.g.,* color, texture, shape) to find other similar studies in a large repository (*e.g.,* picture archiving and communication system, PACS), such as in applications for medical content-based image retrieval. An example of a query is: *retrieve all related patient cases that have nodules with a speckled appearance in the right lower lobe of the lung*. Given the nature of medical images, a visual query-by-example interface is well-suited to the task of query composition: spatial (*e.g.,* right lower lobe of the lung) and morphological attributes (*e.g.,* speckled appearance) are naturally described by a graphical representation. In support of visual querying paradigms, one usability study showed that when asked to specify complex queries, users found visual queries to be more intuitive and expressive than traditional text query languages [115].

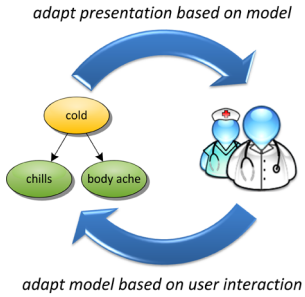


Figure 4.4: VQI's relationship between user interaction and the underlying BBN.

In VQI, the user manipulates a pictographic representation of BBN variables, referred to as *graphical metaphors*. Two types of graphical metaphors exist: 1) a free-hand metaphor that allows the user to sketch a query object (*e.g.,* a tumor) and its environment (*e.g.,* surrounding anatomical structures); and 2) a component metaphor that prompts the user to input numerical or categorical values based on fields in the patient

record. By combining graphical metaphors in different ways, a variety of diagnostic, prognostic, and treatment-related questions may be posed. For imaging-based variables, graphical metaphors take on the properties of their image feature counterparts, allowing users to alter their sizes, locations, relative geometrical positions, and shapes to obtain the desired query. The metaphors bridge a user's knowledge of a familiar domain (*e.g.,* a radiologist's expertise in image interpretation) with an exploratory framework that incorporates variables from other domains (*e.g.*, lab test result). A unique aspect of VQI is that the application guides the user through the query formulation process using the structure of the underlying disease model. The selection of graphical metaphors is context-specific: as the query is built, different metaphors are made available (or removed) to enable the user to draw a logically permissible query. A feedback loop exists between the user and the underlying graphical model, as illustrated in Figure 4.4: given a disease model, contextual information provided by the variables, defined relationships, and user interaction with the model influence what graphical metaphors or functions are the displayed to the user. As the user selects metaphors to formulate a query, the inputs provide some context about the types of variables that are of interest to the user and in turn can be used to identify the subsets of variables in the model that are directly related and relevant for the query. This loop provides a form of relevance feedback: as the user chooses a set of variables to be a part of a query, the system uses this information to refine which metaphors are presented next to the user.

4.2.1. Adaptive interfaces using BBNs

To dynamically adapt the interface, the BBN is used to perform two tasks: 1) capture knowledge about a disease in a probabilistic manner so that inference may be performed by instantiating the model with patient information; and 2) map variables to graphical metaphors and determine when a metaphor is pertinent to the user's query. Attributes of the model are hence used by VQI to determine when a given variable is *relevant* as described subsequently:

- **Variables.** In constructing a disease model, a select number of variables are chosen and modeled to characterize a disease process. Each variable is mapped to a unique graphical metaphor. By way of illustration, an age variable would map to a component graphical metaphor that prompts the user to specify a numerical value. In addition, each variable has a set of states; these states dictate what properties a graphical metaphor can have. For a variable that models the percentage of tumor removed from a patient, the states may be specified by a range of percentage values (*e.g.,* 90-100% resection); the graphical metaphor is responsible for transforming a user's numerical input and assigning it as one of the variable's states. Variable names can also be mapped to a broader knowledge source, such as an ontology, that allows the variable to be defined and placed into the context of other related variables. For example, if a disease model includes the variable word blindness that represents a loss of the patient's ability to read written text, the variable can be mapped to the term *alexia* in the UMLS Metathesaurus and assigned to the

semantic type *T047 - Disease or Symptom*. After mapping all of variables to UMLS, variables with identical or similar semantic types are grouped and presented together in the query interface.

- **Model structure.** The network topology encodes information about the conditional independencies that exist in the model. Based on the Markov assumption, conditional independencies allow the model to be decomposed into smaller subgroups given evidence about certain variables. A variable, given its Markov blanket, can be fully explained and therefore isolated from the rest of the network. VQI leverages this property to identify those subsets of variables in the model related to a given variable of interest. When a variable of interest is selected, VQI examines the variable's Markov blanket to identify additional graphical metaphors to be presented in the interface. Also, the in- and out-degrees of a variable help to determine its relative importance: highly connected variables can be considered more crucial to a disease process than variables that are sparsely connected. Variables that are highly connected are placed in an initial grouping that is presented to the user when no prior metaphors have been selected.

- **Query.** Information about the user's goals can be gleamed from the query itself. The variables that the user selects to be a part of the query elucidate the types of information that the user is seeking from the model. As an example, if the user selects several imaging-related variables, the probability that the user is interested in determining how imaging features affect the outcome of the patient is increased.

Therefore, the model increases the weight of other imaging-related variables in the model so that they are visually highlighted or presented prior to other metaphors in the interface.

The adaptive presentation of relevant graphical metaphors not only simplifies the process of creating a query by reducing visual (selection) clutter, but it also enforces logical rules regarding the order that metaphors are selected to formulate a query. For instance, in neuro-radiology, contrast enhancement, if present, appears around certain image features of a tumor, such as a cyst or necrosis. Therefore, the option to add a rim contrast metaphor is only applicable when a cyst or necrosis metaphor is already present in the query.

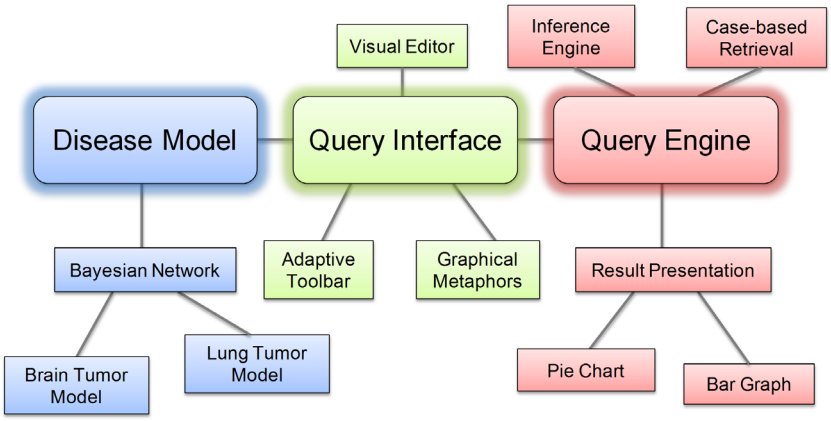


Figure 4.5: A flow diagram showing the components of the visual query interface: disease model, query interface, and query engine.

4.2.2. System framework

The overall system is comprised of three components, as depicted in Figure 4.5 [116]: 1) a disease model specifies the relationships among variables; 2) an adaptive query

interface based on a query-by-example paradigm enables users to pose a query pictorially; and 3) an engine instantiates the model, executes the query, and presents the results.

4.2.2.1. Graphical disease model

Central to this framework is the graphical disease model that quantitatively describes the relationships among variables. The framework is designed to accommodate a variety of probabilistic models that represent a wide range of diseases, but in this instance, the Bayesian belief network of a brain tumor is used.

4.2.2.2. Query interface

The user interface, depicted in Figure 4.6, consists of three panes that enable the user to find available metaphors (adaptive toolbar), pose queries using these metaphors (visual editor), and review/execute queries (query panel).

Adaptive toolbar. The adaptive toolbar (Figure 4.6a) is located at the top of the application and presents available metaphors that can be used in a query. Metaphors are presented based on context: as the user selects structures (*e.g.*, white matter) or metaphors (*e.g.*, edema metaphor) in the visual editor, related metaphors are presented in the toolbar while unrelated metaphors are removed. The determination of whether a metaphor is related (or unrelated) is influenced by the relationships among variables defined in the disease model, as described in Section 4.2.1. Each metaphor is mapped to a corresponding variable in the model. Based on conditional independencies

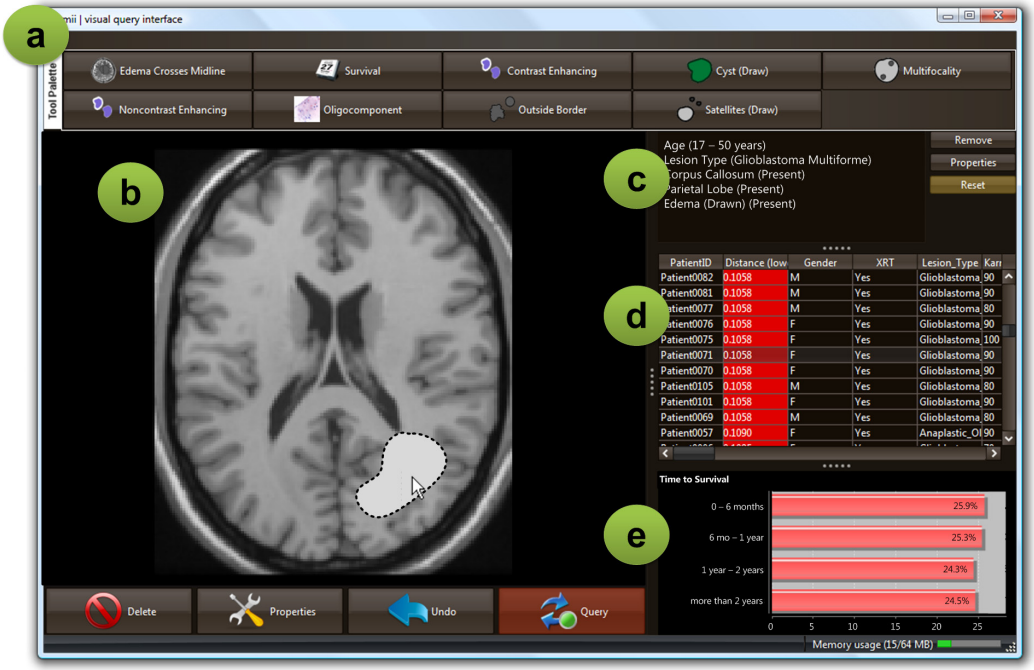


Figure 4.6: A screenshot of VQI. **(a)** The adaptive toolbar displays available graphical metaphors that can be used in a query. **(b)** The visual editor provides space for combining and overlaying metaphors over a labeled atlas to formulate a query visually. **(c)** The query panel translates the visual query into model variables and states. **(d)** The case-based retrieval ranks all patient records in adatabase based on their similarity to the query. **(e)** The results of the prognostic query are displayed as a bar graph showing which state is most probable.

expressed by the structure of the model, subsets of variables and their corresponding metaphors are presented to the user.

Visual editor. The visual editor (Figure 4.6b) is an area in the interface where users pose queries to the disease model using a combination of graphical metaphors. The primary component of the editor is an image viewer that displays slices from a three-dimensional digitized atlas of the region of interest. The labeled atlases provide spatial information about anatomical structures. For example, when the user overlays a tumor metaphor atop a representative slice from an atlas, the anatomical information encoded

in the atlas is used to determine the location of the metaphor and whether the metaphor affects any surrounding structures (*e.g.,* mass effect on the right ventricles).

The user first selects an orientation (axial, sagittal, or coronal) and then chooses a representative image from a set of slices to depict the location of the tumor. The editor provides the user with functionality to select, draw, and customize graphical metaphors. The intent is to allow users to compose and overlay metaphors that pictorially represent a set of image findings on a single, representative slice of the atlas.

The process of posing a visual query is as follows: from a normal or patient imaging study, the user selects a representative slice or location to pose the query; the user iteratively constructs a query by drawing upon the available set of presented metaphors to represent visual features of the disease; and the final query is translated into an object representation that is used to set the states of variables in the BBN as the basis of a maximum a posteriori (MAP) or most probable explanation (MPE) query. Figure 4.7 demonstrates how VQI's adaptive interface works in the context of posing a query in the domain of neuro-oncology: users are presented with a normal brain atlas (ICBM452 [117]), from which axial, coronal, or sagittal slices can be selected. As the user selects metaphors in the visual editor (*e.g.,* contrast enhancement), related metaphors are presented in the toolbar (*e.g.,* define border appearance) while unrelated metaphors are removed (*e.g.,* gender). For instance, when the contrast enhancement metaphor is selected, the user is prompted to define whether the border is thick or thin. A

user progressively composes a visual query, which is automatically translated to values that can be used to instantiate the BBN.

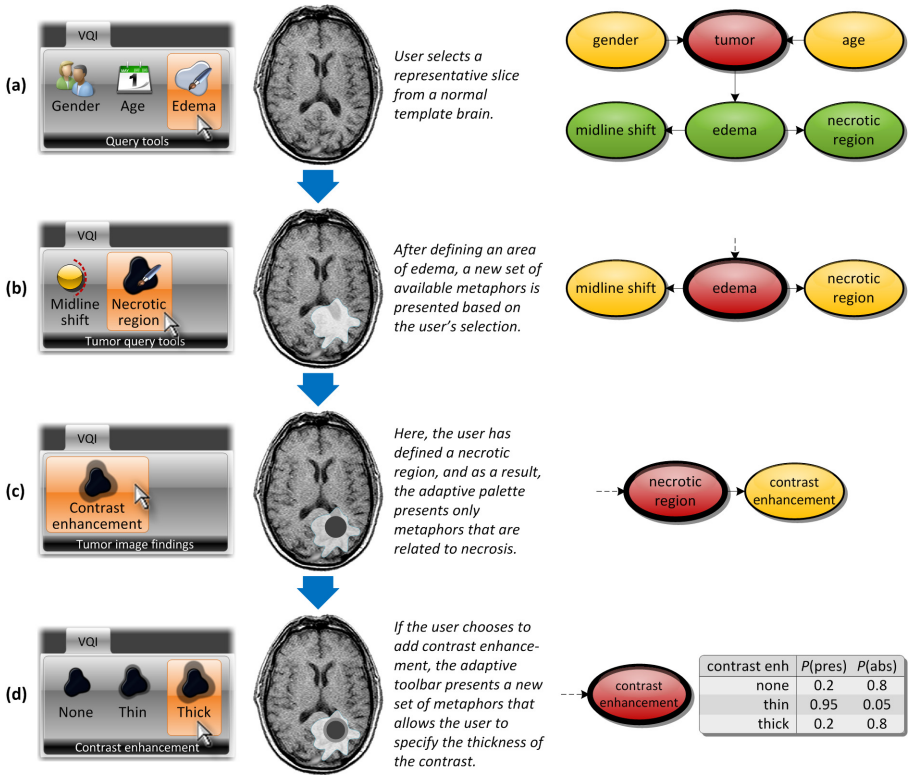


Figure 4.7: Demonstrating query formulation using VQI and how the adaptive interface uses the model to determine the presentation of graphical metaphors [1]. **(a)** The user initially selects a representative slice from an atlas to place a tumor object. **(b)** After drawing an edema metaphor in the query; the model then identifies which metaphors to present next based on the structure of the model. **(c)** After adding a necrotic metaphor, the next relevant metaphor is contrast enhancement. **(d)** The user specifies properties of the contrast enhancement based on the states defined in the variable.

4.2.2.3. Query execution

VQI supports two classes of queries: prognostic reasoning and case-based retrieval.

Prognostic queries. Executing queries against the BBN is accomplished by using an inference engine (*e.g.*, SamIam [118]). A wide range of clinically-relevant questions

can be posed to the disease model using a combination of metaphors. One possible query would be: *what are the states of all other evidence variables in the model, given symptoms of severe edema with noticeable mass effect and midline shift?* (Figure 4.8a) This question is an example of an MPE query where the most probable states of all the variables within the network are calculated. The user can then use this information to pose new queries to determine which combination of treatments can be administered that best improves the patient's outcome given his current state.

Additionally, the user may pose questions that relate to prognosis of a patient given a set of characteristics. An example of a prognostic query is Query 1 posed in Section 3.1, which asks: *what is the most probable range of the Karnofsky Performance status for a 50-60 year old female with a right occipital lobe GBM immediately following complete surgical resection?* (Figure 4.8b) In this query, five metaphors (age, gender, lesion type, location, and resection) are used. First, the user selects the age and gender metaphors to specify the age range and sex of the patient, respectively. Then, the lesion type metaphor is used to specify an area of the brain that is infiltrated by GBM. The system automatically translates the location where the user places the lesion type metaphor using information provided by the atlas; this information is used to instantiate the location variable. Finally, the resection metaphor is used to specify the percentage of the tumor that was removed. Because the query is seeking the most probable state of a particular variable, a MAP query is executed, and the most probable Karnofsky performance status interval is returned.

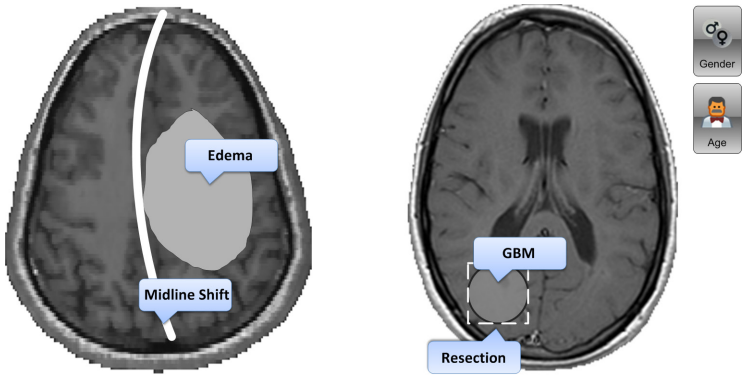


Figure 4.8: Examples of queries that may be posed using VQI. **(a)** A query that depicts a tumor with a severe area of edema that caused a midline shift. **(b)** Another query that utilizes component metaphors to define gender and age and freehand metaphors to define an area of GBM in the right occipital lobe and the area removed during resection.

Case-based retrieval. VQI supports case-based retrieval by using the *Kullback-Leibler (KL) divergence* (D_{KL}) [119]. As introduced in Section 3.2.3.1, the KL divergence assesses the difference between two probability distributions (over the same event space). In VQI, KL divergence is used to measure the similarity between the query and cases in a patient database. Based on the imaging features (*e.g.,* size, location, geometric relationships between objects, etc.) and other non-imaging values specified in the query, the posterior probability distribution for this combination of evidence variables is computed; this value is assigned as $P(x)$. Next, the posterior probability distribution is then calculated for all of the cases in the database using the same variables, but now, each variable is instantiated using the state specified in each case; the resulting value is assigned as $Q(x)$. The KL divergence is iteratively calculated for each case in the database, and the results are ranked from lowest to highest. The case associated with the lowest KL divergence value is the "closest" matching case (with a

KL divergence of 0 being a perfect match). The benefit of using this approach is that unlike traditional case-based approaches, combinations of variables that have not previously been inputted in the database can still be supported: the model will attempt to find the next best combination of features that result in a posterior probability distribution closest to that of the query.

4.2.3. Implementation

VQI is written in Java using the Java Development Kit version 1.6. The anatomical atlases are loaded using the Java Image I/O package using special handlers provided by the LONI Image I/O package to open medical image files (*e.g.*, DICOM). The user interface is implemented using Java Swing with the Substance package providing the look and feel. SamIam [118] is used as the inference engine; it provides an application programming interface that allows external applications to execute various queries (*e.g.*, MAP, MPE) against the model and perform related computations (*e.g.*, calculate KL divergence). Patient case files can be read either from a comma separated file or queried directly from a relational database using a Java database connectivity driver (*e.g.*, MySQL Connector/J). Once the patient data is loaded, the user is presented with an interface for mapping values in the database to variables and states in the disease model. This information is used to index the patient cases.

4.3. Adaptive Electronic Health Record Viewer

The use of the visual dictionary is demonstrated in an application that provides an integrated, longitudinal view of the patient record called *AdaptEHR* (Adaptive Electron-

ic Health Record). AdaptEHR dynamically changes how patient data is presented to the user based on composition rules that are generated based on properties and filters defined in the graphical disease model and other knowledge sources. This section starts by providing motivation for this application, followed by a brief review of related work. It also discusses how AdaptEHR differentiates itself from another context-sensitive visualization, called TimeLine [81]. Finally, basic details on implementation are provided; results from a pilot usability study are provided in Chapter 5.

4.3.1. Background and motivation

Visualization of the patient record is an important part of helping clinicians understand and act on the available data. Clinician tasks can be broadly categorized into three types: 1) what is wrong with the patient; 2) how severe is the problem; and 3) what is the best course of treatment? Answering these questions require a comprehensive understanding of the patient's history and available data types. Visualizations are useful because they transform raw data into visual patterns that are much easier to interpret. A large number of visualizations have been proposed for medical data; Section 2.3 reviews many of these approaches.

This section addresses the issue of combining multiple data elements and visualizations to generate a single display. Such displays provide tools to view and explore the entire patient record from a single interface. However, the sheer amount of data makes displaying all of the available patient data simultaneously impractical to the user. To address this issue, work has been done to generate custom views that display only a

subset of the patient record based on predefined criteria. Zeng [76] and Tange [120] have both explored the area of automatically generating concept-oriented views of medical data. In Zeng's work, a concept-oriented view was generated by using an ontology called Medical Entities Dictionary (MED) [30] to perform concept expansion to identify all data elements in the patient record that is related to particular concept. Relevant concepts were then identified in the patient record. A streamlined page that provided links to departments where related concepts were found was generated; when a user clicked on a particular department, relevant clinical data from the department was shown. Zeng's work is tightly integrated with MED, which provides relational information to link four common medical areas (laboratory, electrocardiography, medical records coding, and pharmacy). On the other hand, AdaptEHR integrates multiple knowledge sources to determine when a data element is relevant to the user: it is not tied to a specific data model. In addition, Zeng describes visualizing the relevant data to fit the look and feel of their institution's electronic medical record. However, pitfalls of that approach are: 1) it is constrained to simple text-based presentation of patient data; and 2) the display cannot display multiple data types simultaneously (*e.g.*, only one type of data can be viewed at a given time). On the other hand, AdaptEHR draws upon a library of visualizations to display information to the user. Multiple visualizations can exist for displaying a single data element: depending on the context, the visualization that best meets the user's information needs and task is selected. My work also attempts to address some of the concerns of concept-oriented organizations of medical data: Bossen [121] argues that such an organization fragments the patient

record across different screens and hence the clinician cannot get a "big picture" of the patient's medical condition. AdaptEHR addresses this issue by displaying patient data through a single interface: users are provided the ability to set filters that determine how much information is displayed based on their information needs. Based on these filters, quantitative values derived from the model's structure and parameters are used to determine whether a given data element is above the threshold and hence, should be displayed.

TimeLine [81] is a problem-centric, time-based visualization of medical data that utilizes a rule-based knowledge base to integrate information from the patient record and presents them to the user based on a set of inclusion rules and available visual metaphors. The goal of the work is to automatically instantiate the display of patient data by providing seamless integration between different heterogeneous clinical data sources, reorganizing the data to create disease- and condition-specific views, and customizing the visual presentation of this data based on user goals and preferences. The system utilizes the International Classification of Diseases (ICD-9) and Medical Subject Headings (MeSH) to provide categories for diseases and associated findings. These diseases then drive the generation of the remainder of the interface: only data elements related to a particular disease is displayed. This dissertation complements and improves upon TimeLine in several ways: 1) relationships between data elements are defined using probabilistic graphical models; 2) unlike ontologies, which encode knowledge as a set of semantic relations, graphical models encode probabilities that

can be used to determine how strongly one variable influences another; this information is used to alter the appearance of data elements (*e.g.*, size, proximity, layering); 3) the inclusion rules originally proposed in TimeLine have been augmented to include information provided by the disease model; and 4) user input is now taken into account to dynamically determine what additional information is presented on the screen. AdaptEHR is not meant to be a replacement of TimeLine; rather, it is a generic visualization framework that can be used to generate interfaces such as TimeLine.

4.3.2. System description

Three components have grown out of the development of AdaptEHR, facilitating different aspects of the system: 1) an *information extraction* component identifies data in the medical record that are relevant to a condition's diagnosis and treatment; 2) a *prioritization and relation* component that uses the graphical disease model to link and rank data elements based on how relevant they are to a given query; and 3) a *visualization* component that selects the appropriate visual metaphor for the data being displayed and how the data is laid out on the screen. To ground the discussion, I first introduce the visualization component to provide context for understanding the other parts of AdaptEHR that are used to change the display.

4.3.2.1. Display

The user interface for AdaptEHR, depicted in Figure 4.9, is comprised of three components:

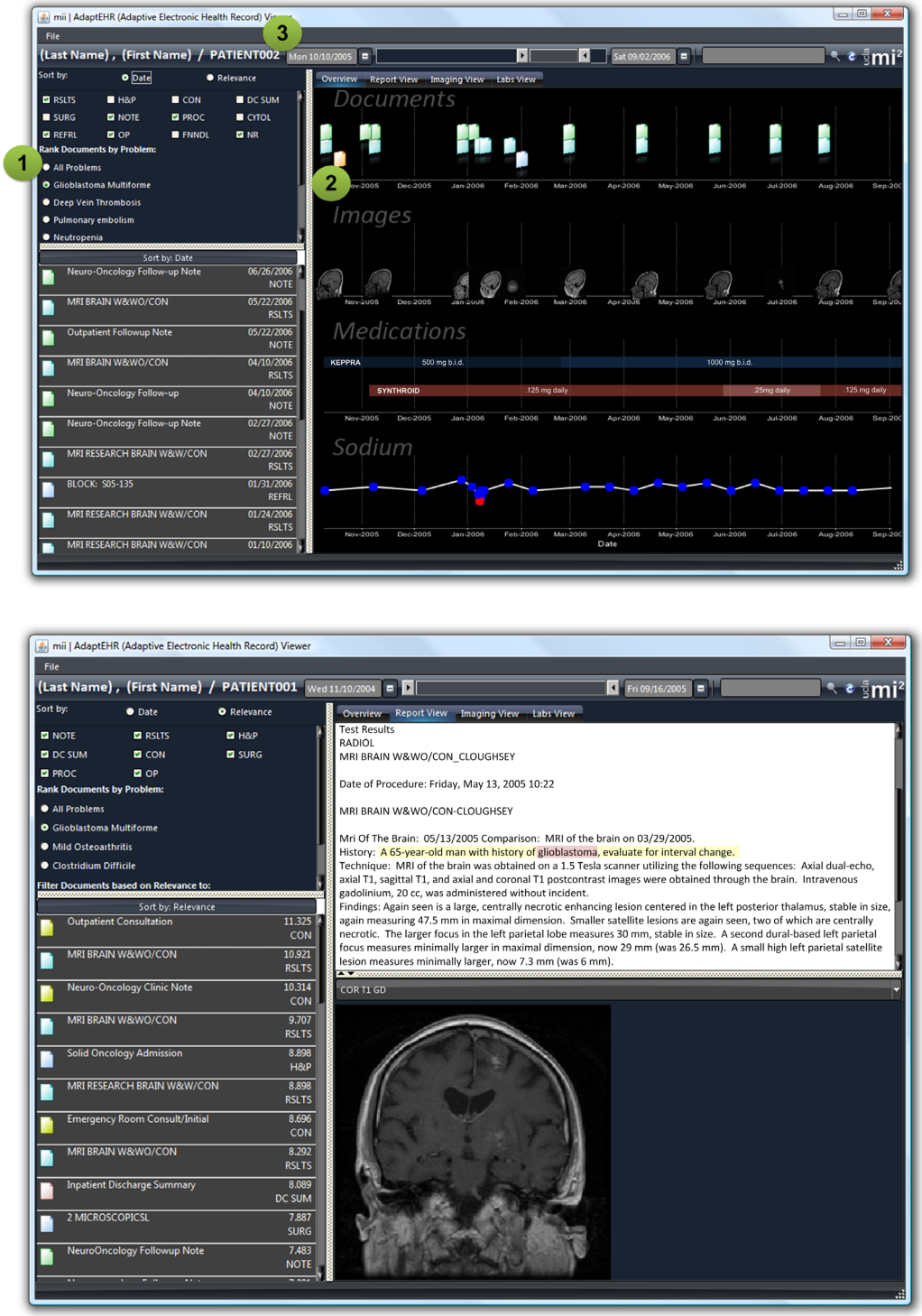


Figure 4.9: Screenshots of AdaptEHR. **(a)** The main interface is comprised of 1) the filter panel, 2) timelines, and 3) temporal filter. **(b)** The data viewer provides a detailed view of the patient data.

Data filters. The left side of the display (Figure 4.9a-1) presents all of the configuration options that may be used by the user to customize the interface using the filters enumerated in Section 4.3.2.3. Users can select the necessary checkboxes, radio buttons, and sliders to change the types of information displayed in the other parts of the interface.

Timelines. Time is delineated along the horizontal (x-axis) with months and years marked off accordingly. The timeline display acts as a canvas, allowing graphical elements corresponding to different pieces of information from the patient record to be overlaid. Upon opening a patient's record, three timelines are initially presented: the documentation history, imaging history, and encounter history. Selecting a problem from the filter panel adds additional timelines to this view based on the problem selected, data available, and user. These visual cues are modeled after the TimeLine system presented in [122]. Events representing patient data are denoted along the timelines by iconic links:

- Document icons are color-coded by source or data type.
- Quantitative clinical lab values are plotted; the normal value is indicated by blue points while abnormal values are plotted in red.
- Medications and treatments are visualized by bars that span the duration of the event. Interventions are visualized as icons that convey the type of intervention.
- Imaging procedures are represented by thumbnail images indicative of the exam.

Data viewer. Data in the patient record is accessed through the timeline display. Once a user selects a specific data element (*e.g.*, an icon for a particular document), the raw data from the patient record is loaded in a generic viewer that allows users to examine documents, images, and labs in more detail. The viewer can be split either vertically or horizontally to allow comparisons between the same data type or to view different data types that relate to one another (*e.g.*, medical images and their associated radiology report). Different visualizations (*e.g.*, point-based, interval-based) are used to present the data based on the user model and user-specified filters. The visual dictionary selects a visualization to use given a data element and its context.

Interaction with AdaptEHR can be summarized in the following steps:

1. The user selects a particular medical problem that has been mined from the patient record.
2. The corresponding knowledge bases (*e.g.*, disease model) for the selected condition are loaded.
3. The visual dictionary is initialized with seed variables identified based on the user/task models, and the initial display is generated.
4. The user interacts with the model by specifying a set of filters provided.
5. The filters are translated to a set of associated variables and states that are used to instantiate the model.

6. Based on variables that the user specifies as part of the query, influential findings and paths of influence are identified.

7. The system determines the highest ranking documents and data elements and displays the data using appropriate visual metaphors.

8. An updated display is generated.

9. As the user continues to interact with the system, steps 4-8 are repeated, adding and removing metaphors from the display depending on their relevance to the user defined concept.

4.3.2.2. Information extraction

Information extraction is the process of identifying and extracting relevant events, entities, and relationships from unstructured clinical data that have significance to patient care. AdaptEHR works primarily with three types of medical data: clinical documents, medical images, and laboratory values; methods for characterizing them are discussed in the following sections.

Clinical documents. Unstructured free-text documents contain large amounts of useful medical information; however, free-text is not amenable to analysis by a computer. A natural language processing (NLP) system [123] is used to identify relevant biomedical concepts from free text and perform semantic interpretation to associate attributes to related concepts. Concepts that are extracted include:

- Temporal concepts. Time is an essential component to making an accurate and complete clinical diagnosis. The date of when a particular problem or finding is present conveys temporal information that provides historical context for a given medical problem. Clinicians would be able to easily determine whether a particular disease is a recurrence or a newly discovered problem, all in relation to the other events in the patient's medical history.

- Spatial concepts. Anatomic (spatial) descriptions of findings are fundamental to disease understanding, as symptoms are often the result of changes caused by the disease to surrounding regions. Mapping anatomical descriptions extracted by NLP from clinical documents to spatial representations (*e.g.*, standardized atlases/anatomical reference frames) provide both improved visual depiction of how the problems are distributed in a patient and a common reference frame for facilitating spatial reasoning related to patient outcomes. Anatomical phrases identified in the patient record are mapped to standardized concepts found in controlled vocabularies.

- Existential concepts. In medical reporting, physicians often qualitatively assess a level of certainty for the existence of a given problem. Existence describes whether a problem is observed in the patient at a given time. Existential values may include: *definitely exist*, *likely*, *possibly*, *less likely*, *cannot be ruled out*, *no evidence of*, and *does not exist*. These values can also be examined globally across various documents to determine how an instance of a problem relates to other occurrences.

For example, given that a problem is mentioned in several clinical documents, each instance may be labeled as one of four categories: new, recurrent, old, or resolved. Existence of a problem or symptom implies the existence of a disease or conversely, its resolution.

- Causal concepts. The notion of "causality" and clinical medicine are inherently intertwined. Patient care is driven by causal considerations: symptoms manifest due to underlying etiology, which in turn are the result of some (abnormal) biological phenomena. An important aspect of organizing clinical information is to capture the cause-effect relationships among variables of interest, such as treatments, exposures, preconditions, and outcomes. In conjunction with existential information, causality is the basis by which a physician can determine the existence of a disease: diagnosis is a conclusion that is arrived at by analyzing the symptoms/problems of a patient. NLP may be used to identify causal links between concepts (*e.g.*, problems to findings, medications to disease response); this information is used to influence how data elements are positioned on the screen (*e.g.*, causally related concepts are positioned close together).

The input to an NLP system is an unstructured body of a clinical document that is written by a physician; the output is logical frames that identify concepts within the text and their associated attributes. An example of a frame-based output is shown in Figure 4.10 for the sentence: *there is a large well-circumscribed 5cm mass in the left upper lobe consistent with adenocarcinoma.* The frame representation enables several

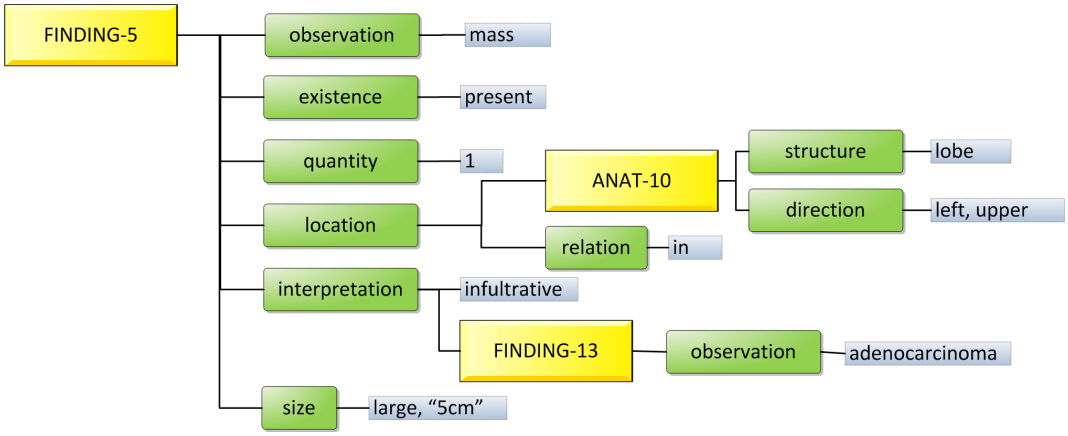


Figure 4.10: An example of a frame output for the sentence: *there is a large well-circumscribed 5cm mass in the left upper lobe consistent with adenocarcinoma*.

types of manipulations: 1) the representation facilitates mapping relevant concepts in a document to a disease model; 2) specific attributes that modify a particular concept are used to instantiate the appropriate state in the model; and 3) the extracted attributes can be translated into a visual summary thus eliminating the need for users to browse through entire documents to find relevant information.

Medical images. Methods for image understanding are necessary to make sense of the pixel data and identify any relevant features using texture, color, and shape analysis. In addition, metadata provided by the header file can be used to help categorize images and relate them to documents. AdaptEHR utilizes a subset of tags and elements, which are presented in Table 4.3 to help categorize and associate imaging data with other parts of the patient record. The tags provide context about how the image was acquired and the content that the image depicts. The study and series descriptions provide an idea of the sequence used; the body part examined provides information about

Tag	Name	Purpose
0008,0032	Acquisition Time	Used to generate time-oriented view
0008,0050	Accession number	Used to link DICOM image series with available radiology reports
0008,0060	Modality	Used to filter images by modality
0008,1030	Study Description	Used to filter studies based on keywords in field
0008,103E	Series Description	Used to filter images based on keywords in field
0018,0015	Body Part Examined	Used to filter images based on anatomical region based on keywords provided in field

Table 4.3: A summary of the DICOM header fields that are extracted and used to filter medical images.

the anatomical region that is depicted. The accession number is used to link images to related radiology reports that were dictated by a radiologist; corresponding images and reports can be retrieved and displayed at the same time. Medical images are visualized in AdaptEHR using one of the following ways:

- Thumbnail. In the overview display, medical images are represented by a thumbnail image that provides a snapshot of the type of data contained within the imaging study.

- Context-in-detail. Users who desire an overview of the entire imaging study can use the context-in-detail view, which displays medical images in a film-strip viewer. Individual slices are displayed in a grid; users can scroll through the images or hover over a single image and view a larger version of the selected slice.

- Detailed view. For users who wish to manipulate the image (*e.g.*, add annotations, change window/level), clicking on a specific image will initialize a fully functional image viewer that provides a host of functionality to modify and annotate the image. Different imaging studies for a given medical problem can be viewed simul-

taneously, allowing the user to view changes in image features over time. Such comparison would be useful in evaluating whether a particular treatment is effective.

Laboratory values. Laboratory data is typically organized hierarchically in a generalization-specialization manner as described in Section 3.5.1.3. In AdaptEHR, this hierarchical organization is visualized using a GUI tree component that allows the user to "drill down" to a group of tests or the specific test of interest. When selecting individual test results, each result can be visualized in four ways:

1. Tabular display. Lab values are displayed numerically in a table. Abnormal values are highlighted in red. The degree of abnormality (number of standard deviations from the normal range) is shown by coloring the cell with varying shades of red.

2. Graph display. Lab values are plotted on a time series graph. Values that are abnormal are colored in red. The rate of change between each data point is calculated; the stroke color changes based on the calculated slope. Sudden changes in test results in a short period of time can be denoted with a vibrant color while progressive changes are presented using more muted colors.

3. Mixed display. This presentation format combines tabular and graph presentations in a single display. The interfaces are linked: clicking on a specific cell in the tabular display highlights the corresponding point in the time series plot.

4. Radar chart. While the aforementioned visualizations are designed for a detailed view of trends in the data, the radar chart is a way of summarizing changes between normal and abnormal lab values. As first presented in [47], the radar chart allows users to compare lab results using the shape of the chart. For instance, a panel with five tests can be represented as a pentagon if all of the test results are normal. However, when any of the test results are abnormal, the pentagon becomes skewed based on how abnormal the test result is. This visualization provides a quick method for determining which panels among all laboratory tests have abnormal results.

In general, the AdaptEHR GUI follows the "overview, zoom and filter, and details on demand" approach [112]. Users are initially presented with an overview abstraction of the data: data elements are presented using icons or thumbnails. As the user selects individual elements or poses a query to the system, the user interface dynamically changes to reveal additional details within the patient record that are relevant to the user's query.

4.3.2.3. Prioritization and relation

One of the primary features of AdaptEHR is its ability to prioritize and relate data elements in the patient record based on the relationships defined in a graphical disease model or ontology. This section describes several of the key filters that permit users to change the criteria by which data is filtered and prioritized on the screen. These filters specify the parameters that are used to instantiate the composition rules; hence, vary-

ing the settings on these filters affects how patient data is displayed. Five different types of filters have been developed:

Problem filter. The problem list is the primary filter that determines which data elements are presented to the user. A patient's problem list can be generated by: 1) using billing codes (*e.g.*, ICD-9) that are assigned to each data element for billing purposes; or 2) using automated methods to extract reported conditions from medical reports (*e.g.*, discharge summaries) such as NLP [124, 125]. Selecting one or more conditions invokes all knowledge sources that have information about the selected problem: graphical disease models and medical ontologies are analyzed to determine the types of data elements that are relevant to the selected problem; these methods are discussed in Chapter 3. Given this information, all of the data elements in the patient record are presented using one of the following methods:

- Relevant clinical documents are ranked and displayed on a timeline interface. Ranking is performed using the tf-idf weighting method that is discussed in Section 3.5.1.1.

- Thumbnails of relevant medical imaging studies are rendered on the timeline. For radiologists, selecting a thumbnail loads up the entire image either along with the radiology report or with a set of manipulation tools (*e.g.*, window/level, magnification) that can be used to further explore the image. For all other users, selecting a thumbnail will load the context-in-detail view, providing the user with a quick way to view all of slices in the study.

- A list of medication, interventions, and laboratory tests that are related to the problem is overlaid and displayed on the timeline.

	Primary Care Physician	Radiologist	Patient
Clinical data			
Demographics	•	•	
Medical history	•	•	
Vitals (BP, BMI)	•		
AST, ALT, liver	•		•
Total cholesterol	•		•
HBV-5 panel	•		•
Imaging (abdominal)	•	•	
PCP reports	•	•	•
Hepatology reports	•	•	•
Radiology reports	•	•	
Medication history	•	•	•
Prioritization	1. Labs 2. Medication history 3. Medical history	1. Imaging 2. Radiology report 3. Hepatology report	1. Labs 2. Medication history 3. PCP reports

Table 4.4: A table depicting the information that is contained in a user model for AdaptEHR.

User filter. As discussed in Section 3.4.2, user models play an important role in providing information about the user's information needs and preferences. A simple rule-based user model has been implemented in AdaptEHR that specifies: 1) which data types are presented to the user and 2) how data is prioritized. Table 4.4 illustrates the types of information that are stored in the user model. All of the data types found in a patient's record are enumerated in the model; for each user group, the model specifies when a data type is to be presented. For example, demographics would be shown to both the primary care physician (PCP) and radiologist but would not be presented to the patient. In addition, each data type is also prioritized based on how important the data is to a user. For instance, laboratory results are the most important to both the PCP and patient, but on the other hand, imaging studies are more important to the radiologist.

Temporal filter. Concepts and data elements can be filtered based on when the event occurred or based on a time stamp respectively. Dates are important for determining patterns that may occur over the course of a chronic illness or treatment. Patients with large patient records (*e.g.*, data documented over a long period of time) can be filtered so that only a small portion of the data is displayed. The temporal filter affects all views in AdaptEHR; setting the temporal filter removes any data elements that do not occur within the specified time period from the display.

Semantic filter. These filters define the types of information that are used to rank patient data. Semantic filters are driven by the groupings described in Section 3.2.1.2; they are used to rank data elements based on the subsets of patient data that are created by these groupings. For example, when *Therapeutics* is selected from a list of semantic groups, only variables that map to terms in MeSH with the group heading *Therapeutics* (*e.g.*, chemotherapy drugs, surgical procedures) are used to rank the documents. In [126], the authors propose fifteen high-level semantic groups that reduce the conceptual complexity of the large domain covered by the UMLS; AdaptEHR leverages this work to cluster individual semantic types defined in the UMLS to broader semantic groups. For example, given that a disease model contains variables such as necrosis, headache, Carboplatin, and entire left ventricle, these variables can be categorized into disorders (necrosis, headache), medications (Carboplatin), and anatomy (entire left ventricle).

Relational filter. The final filter exploits the probabilistic information encoded in the disease model. Unlike the other filters, which could be driven by an ontology or rule-based knowledge source, the relational filter utilizes the probabilities to weigh how strongly variables are related to one another. Data elements are then reorganized based on how strongly they influence one another. Influence among data elements is quantified by computing measures such as the strength of influence, value of information, path of influence, or influence of evidence. Visually, this filter is represented as a slider bar UI component, which allows users to specify a threshold value. Any data element whose relationship is determined to be below the set threshold from a given target variable is removed from the display. Filtering occurs dynamically; as the user slides the value to higher or lower thresholds, data elements are added or removed. In addition, the user can specify whether to lay out the data based on how strongly related data elements are to each other. For example, using the spring-embedded layout discussed in Section 3.5.2.3, visualizations are placed closer or farther from each other based on the spring constants assigned to each edge. Variables that have larger strengths of influence will have a higher spring constant (therefore displayed closer together) while less-related data elements are displayed further apart (repelled).

4.3.3. Implementation

AdaptEHR is written in Java using Java SDK 1.6. The user interface is built using Java Swing; the look and feel is provided by the Substance package. The SwingX pack-

age provides several advanced UI components such as date selection, custom painters, and transparent panels.

Clinical documents are characterized using the MII NLP toolkit [127]. The system takes a sentence as input and produces a set of logical frames based on a rule-based semantic interpreter, which is modeled for the radiology domain. The system has previously been applied towards the automated generation of a patient problem list [125]. Medical images are rendered using an image viewer package [128] that is capable of reading both DICOM and Analyze data formats and outputting their header information. The image viewer is integrated into AdaptEHR and provides the advanced manipulation functionalities such as zoom/pan, window/level, annotations, and image layout. Charts (*e.g.*, timelines, time series, bar graphs, radar plots) are generated using the JFreeChart package.

To create the visual metaphors, various packages are used. The prefuse toolkit [32] is used for graphs and trees. Tables are generated using the default Java table component and the GlazedLists package to provide methods for sorting columns, highlighting rows, and dynamically filtering content. Calculating the strength of influence and value of information is performed using SMILE, a programming interface for the inference engine GeNIe [129]. Indexing of the text, such as concepts extracted from clinical documents and header files, is done using an open source search engine called Lucene.

4.3.4. Example Query

One of the contributions of AdaptEHR is its ability to prioritize and relate data elements in the patient record based on the relationships defined in a graphical disease model or ontology. The result is the ability to filter and rank data based on constraints defined by the user's query. To provide an example of how AdaptEHR adaptively displays patient information based on a query, I revisit Query 8 that was posed in Section 3.1: *is my non-small cell lung cancer patient eligible to participate in a study that compares erlotinib to standard chemotherapy?*

To be eligible, the patient must have either Stage IIIB or IV non-small cell lung cancer, have an ECOG performance status of 2 or less, Karnofsky Performance Status of greater than 70%, serum calcium < 12 mg/dl, and a clinically or radiologically measurable disease based on Response Evaluation Criteria in Solid Tumors (RECIST). Patients must not have existing gastro-intestinal abnormalities, any concurrent anti-cancer therapy, prior treatment with EGFR inhibitors, other active malignancies, brain metastases, several abnormalities of the cornea, or significant cardiac disease.

To answer this query, the system needs to have prior information about the eligibility requirements and how they relate to the overall disease of lung cancer. If a model that includes variables representing different aspects of the eligibility criteria exists, it can be used to obtain information about how variables relate to one another and in the context of the disease as a whole. AdaptEHR uses the provided information to dynamically present the patient data by highlighting the parts of the patient record that pertains

specifically to these criteria. For example, any mention of the patient's lung cancer stage, ECOG performance status, and past treatments are extracted and highlighted since they map to corresponding variables in the model. In addition, any CT or MR imaging studies that have been acquired that are related to the patient's lung cancer are also displayed; this information is used to determine whether the patient has a measurable disease based on RECIST. The treatments panel shows the duration and dosages of chemotherapy that patient has received, if any. Finally, the laboratory test panel displays recent measurements of serum calcium; if the patient had any measurements that were above the eligibility criteria (> 12mg/dl), those values would be highlighted in red. The display is depicted in Figure 4.11.

Figure 4.11: Screenshot of the AdaptEHR interface showing how the display is tailored to answer Query 8. **(a)** User specifies the context used to tailor the display. **(b)** The selected filters are used to instantiate the composition rules. **(c)** The display highlights concepts in documents that are relevant to the eligibility criteria. Phrases with terms that match variables in the model are extracted and presented to the user. All other documents are rendered transparently. **(d)** Related images are enlarged and highlighted. **(e)** Information regarding the patient's past treatments is extracted and highlighted.

CHAPTER 5

Evaluation

5. Overview

A goal of this dissertation is to use graphical disease models as a way to generate context-sensitive visualizations of patient data. In preceding chapters, various approaches have been described for using properties of a graphical disease model and other knowledge sources to determine whether elements in the patient record are relevant and how they are presented to a user. In this chapter, these approaches are evaluated using a pilot study to ascertain whether they meet the original goals of this project. Similar works, such as Zeng et al., have presented methodologies that measure precision and recall for identifying relevant patient information [97]. The evaluation of this work is described in two parts: first, components are assessed quantitatively using measures such as precision, recall, and information reduction to gauge their efficacy; second, user acceptance is appraised by soliciting opinions from a target group of users through a usability survey after each user has completed a series of tasks using VQI or AdaptEHR (Chapter 4). The organization of this chapter generally follows that of [130], which divides the task of evaluation into three parts: verification (Section 5.1), validation (Section 5.2), and assessment of human factors (Section 5.3).

5.1. Test data

The evaluations were performed using de-identified patient medical data that have been acquired from the UCLA Ronald Reagan Medical Center with prior institutional review board (IRB) approval. Patient data was acquired from the hospital information system (HIS) to a research database using a web-based application framework called DataServer [41]. Available data included radiology reports, discharge summaries, surgical reports, radiological imaging studies, chemistry results, encounter information, microbiology, blood, and pathology findings. Protected health information as defined by the Health Insurance Portability and Accountability Act (HIPAA) was automatically removed from the data as part of the acquisition process. Results of de-identification were manually validated for all patient cases that were presented to test participants during the usability studies.

5.2. Verification

Verification is the process of testing whether individual components of the system have been implemented correctly and perform as expected. In this work, two components are evaluated: 1) the accuracy of the underlying graphical disease model, and 2) the process of mapping variables in the disease model to concepts represented in a medical ontology.

5.2.1. Model validation

A key factor that determines the utility of this work is the accuracy of the underlying graphical model in representing knowledge about a disease. Historically, BBNs have been evaluated by measuring their predictive power. First, a set of test cases is compiled and used as ground truth. Then, after the BBN has been trained, values from the ground truth are used to instantiate the model and predict the value of a target variable. Results provided by the model are compared with the known answers to determine the accuracy of the model. A second approach gauges the impact of a disease model in providing decision support to target users. Study participants are first asked to provide answers to a set of clinical queries drawing solely upon their knowledge and experience. Next, they are given an opportunity to change their answers based on information provided by the model. The user makes a final judgment whether or not to change their original answers based on the new information. This test measures any improvement in diagnostic accuracy that the user experiences when aided by the model. Finally, model performance can be compared to other types of diagnostic/prognostic models that represent a baseline approach, such as a decision tree or logistic regression model. Using the aforementioned techniques, I discuss the evaluation of the disease models used in my work.

Glioblastoma multiforme (GBM) model. A ten-fold cross validation was performed to measure the accuracy of the prognostic GBM model. The entire dataset of cases (200 patients) was divided into ten equal parts; nine of the ten parts (180 cases) were

used for training, and the remaining part (20 cases) was used for testing. This process was repeated ten times by swapping the part that was held out for testing; the overall accuracy was computed by averaging the results across all ten tests. The prognostic model achieved an average accuracy of 72% when trying to predict the variable time to survival (TTS). To provide a baseline comparison, a stepwise multivariate logistic regression model was created using the same set of variables as the BBN; it predicted TTS with an accuracy of 86%. Several factors may explain why the baseline model performed better than the BBN: 1) in the baseline model, every variable is represented using only two states underscoring the sensitivity of the BBN model to discretization, and 2) the BBN model is a static model that does not represent how the disease varies over time. Another limitation of the current study is the limited patient population. Future work would include: 1) experimenting with different discretization technique, and 2) generating a model that draws upon data from multiple institutions, which would the address the issues of sample size and bias.

Imaging-centric brain tumor model. Similar to the GBM model, a ten-fold cross validation study was performed to measure the predictive power of the imaging-centric brain tumor model. A different subset of 152 patient cases was identified from the neuro-oncology database and used to train and test the model. This subset of patients was selected because they contained a variety of brain tumor cases outside of GBM, and they had the most complete information about imaging features. Each patient case was tested with a set of queries. Sample queries included (with the model's

performance for that query in parenthesis): 1) *given the age, gender, and lesion type, what is the most likely values for necrosis and lesion size* (60.2% accuracy); 2) *given the MR findings of edema, lesion size, and necrosis, what is the most likely lesion type* (75.66% accuracy); and 3) *given the MR findings of edema, lesion size, and necrosis, compute the most likely scenario for the remaining network nodes* (62.91% accuracy). A key issue of this model was the sparseness of the conditional probability tables due to the limited patient population used to generate the model; as in the GBM model, this model would benefit from an increased study population size that is drawn from multiple institutions.

5.2.2. Mapping variables to an ontology

The first experiment measured the completeness of mapping variables in disease models to an existing medical knowledge source (*e.g.*, Unified Medical Language System, UMLS). In [131], the authors extracted 2,268 distinct concepts from 24 clinical documents using a custom noun phrase detection algorithm. They attempted to map extracted concepts to the 1999 UMLS Metathesaurus resulting in the successful mapping of 76% of the concepts. In this study, a similar experiment was attempted using variable names rather than terms extracted from clinical documents. This experiment sought to answer the following questions:

1. What is the percentage of variable names that are successfully mapped to concepts in the UMLS?

2. What types of variable names are not mapped successfully?

Variable names were parsed from the disease models and mapped to UMLS using MetaMap Transfer application [132]. The GBM model contained a total of 17 variables; the imaging-centric brain tumor model had 30 variables; and the liver disorder model contained 71 variables. The results of mapping are reported in Table 5.1.

Model	Matched Variables (Percentage)	Unmapped Variables
GBM Model	14 / 17 (82.4%)	Time to survival
		Time to progression
		Overall tumor status
Imaging-centric brain tumor model	24 / 30 (80%)	Edema crosses midline
		Rim contrast
		Tumor crosses midline
		Contrast enhancement
		Noncontrast enhancement
		Time to survival
		Outside edge
HEPAR II	68 / 71 (95.7%)	Hepatotoxic medications
		Total triglycerides
		Yellowing of the skin

Table 5.1: A listing of variables that do not have matching concepts in the UMLS.

On average, MetaMap Transfer was able to find the correct concept as one of the top candidates for 86% of the variables. However, the system was not able to perform fully automated concept matching; manual inspection was still needed to identify which of the top candidates was indeed the correct match for a given variable. Additionally, while the coverage of terms in the UMLS has improved since [131] was published, improvements in representing variants for each unique concept and capturing addi-

tional concepts from specialized domains are still needed. Variables that were not mapped to a concept may be grouped into two categories:

- **Phrases.** Matching term variants to concepts remain a challenge. In particular, phrases used to describe a phenomenon are often difficult to map to a specific term. For instance, while the phrase *yellowing of the skin* cannot be found in the Metathesaurus, the term *yellow skin* is found and is mapped to the term *jaundice*. Users cannot expect the UMLS to include every type of variant that may exist to describe a specific concept; one solution would be to standardize the way variables are named based on a common set of data elements.

- **Domain-specific terms.** The 2009AA release of the UMLS Metathesaurus incorporates 129 knowledge sources that represent terms from different domains and in various languages. Despite the growing coverage, some domains are still poorly represented. Other medical lexicons may be used to supplement existing UMLS sources. For example, the term *time to progression* is not represented in the UMLS but is represented in the NCI Metathesaurus (NCI-M). The NCI-M is an extension of the UMLS Metathesaurus that is tailored to meet the needs of the cancer research community by excluding irrelevant terminologies and adding others particular to biological and oncological research. Other sources of terms include the Foundational Model of Anatomy (for anatomical and spatial terms), RadLex (for radiological terms), and Gene Ontology (for genomic terms).

Improvements in the mapping process can be made by: 1) standardizing how variables are represented in the model by utilizing a set of common data elements, and 2) improving the algorithms that match phrases with concepts in the medical ontology. Common data elements (CDEs) have grown increasingly popular as a way to integrate data acquired across multiple clinical trial sites [133]. The NCI has spearheaded efforts to develop a dictionary of CDEs that clearly define: 1) how data elements are named; 2) how data elements are collected; 3) the values and states that each data element can be assigned; and 4) the data type. However, creation of CDEs requires a team of domain experts to decide on each of the four attributes; while this process is time consuming, as progress is made on making new CDEs available, CDEs can be applied towards standardizing the representation of variables in a disease model and facilitating the creation of models that utilize data from multiple institutions. The second improvement addresses the need for different algorithms that map phrases to UMLS concepts. While this dissertation utilizes MetaMap Transfer, which is a popular tool for performing the mapping task, alternative approaches have been proposed to address the perceived performance issues of this tool: [134] provides a reference on different approaches for word sense disambiguation. [135] presents a concept mapping application that creates a of consistent concepts graph using relations defined in the Semantic Network; the graph is then used to infer how ambiguous terms should be mapped to concepts.

5.3. Validation

Validation determines whether the implemented framework meets the original design objectives. As outlined in Chapter 1, the goals of context-sensitive visualization are to assist users with identifying relevant data elements in the patient record given some context (*e.g.*, medical problem, task) and reducing the amount of patient information that is displayed. This section describes the design and execution of two experiments that measure recall, precision, and percentage of information reduction.

5.3.1. Gold standard

To evaluate the system's ability to identify relevant documents, a test set of 1,299 patient documents was created. The documents were retrieved from the HIS using the process described in Section 5.1. Documents were primarily composed of radiology reports that had been annotated with an ICD-9-coded diagnosis. A subset of documents that specifically pertained to brain tumors (all problems that are children of ICD-9 subgroup 191) and cirrhosis (571.x) were identified. However, in multiple instances, documents that may have pertained to brain tumor or cirrhosis were not labeled as such. This discrepancy was due to the fact that the ICD-9 codes for each document were sometimes assigned based on the primary diagnosis of a patient, not by the content of individual documents. As a result, all of the documents were manually inspected to ensure that the assigned ICD-9 code correctly reflected the contents of the document. A total of 50 documents pertained to brain tumors; 26 documents pertained to cirrhosis in the test set.

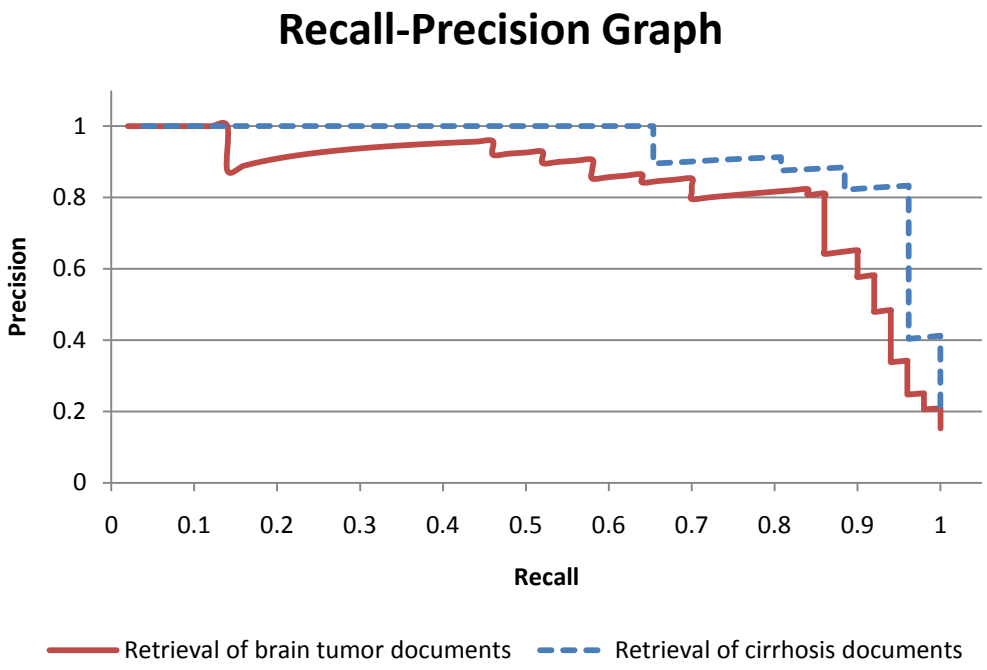


Figure 5.1: A recall-precision graph that summarizes the results of ranked retrieval for cirrhosis-related and brain tumor-related documents from a large corpus of 1,299 documents.

5.3.2. Information retrieval

First, variables in the disease model were mapped to concepts in the UMLS Metathesaurus. Query expansion was used to find concepts in the Metathesaurus that were synonyms, parents, children, and siblings of each variable. The expanded set of terms was then employed to search the entire document corpus. The number of matches for each term was saved; a weight for each variable using tf-idf was computed from these values. To rank documents, the weights of each variable were summed together for each document. Documents that had higher weights were considered to be more relevant to a given context than ones with lower weights. To compute the recall and preci-

sion of the ranked results, the ICD-9 annotations were used to determine how many of the top ranked results were indeed classified as being relevant to the disease of interest.

Unlike unranked retrieval which measures an algorithm using a single set of recall, precision and F-score values, ranked retrieval requires the definition of a spectrum of relevance. Hence, precision is measured at different levels of recall and is best presented in a recall-precision graph (Figure 5.1). The graph illustrates that the approach performs well: of the 50 documents that pertained to brain tumor in the test set, 41 of them were ranked in the top 50 documents. In addition, 25 of the 26 documents pertaining to cirrshosis were returned in the top 30 results of the retrieval. The experiment validates that the process of query expansion can be used to identify relevant documents in the patient record. In this experiment, all of the variables in the model were used to rank the documents. A future test could be done to determine whether documents can be accurately ranked given only a subset of the variables in a model. For example, if the user is specifically interested in viewing documents that relate to *Signs and Symptoms* associated with cirrhosis, then only the subset of variables in the liver disorder model that are associated that semantic type are selected, expanded, and used to search the patient data. The test would determine whether this approach is sensitive enough to identify documents with desired content based on the selected variables.

Test Name	LOINC	Description
ALT	1742-6	Measure of cell liver death from inflammation.
AST	1920-8	Measure of cell liver death from inflammation.
GGTP	2324-2	Elevated activity indicates abnormal bile flow or bile duct disease.
Alkaline Phosphatase	6768-6	Elevated activity indicates abnormal bile flow or bile duct disease.
Total Bilirubin	1975-2	Elevated levels are proportional to the amount of liver dysfunction.
Albumin	1751-7	Decreased levels are an indicator of cirrhosis.
INR	6301-6	Related to Prothrombin Time (PT), blood clotting, indicator of liver function.
Amylase	1798-8	Lower values indicate liver dysfunction; related to increases in bilirubin levels.
Cholesterol	2093-3	Elevated levels implicated in fatty liver disease.
Triglycerides	2571-8	Elevated levels implicated in fatty liver disease.
ESR	18184-2	Measures inflammation, indicator of liver disease.
Hepatitis Panel (HA-Ab-IgM)	13950-1	Indicate recent infection with hepatitis A virus.
Platelet Count	13056-7	Lower values are measured typically after the onset of cirrhosis.

Table 5.2: Laboratory tests represented in the diagnostic liver disorder model.

5.3.3. Information reduction

The second validation test determined the degree of reduction of the amount of information that was presented to the user using the knowledge sources. The evaluation procedure followed one that was performed in [97]; the percentage of information reduction was calculated by comparing the amount of relevant information determined using the graphical disease model and the amount of information available in the patient record. In this evaluation, only laboratory values were considered. The liver disease model described in Section 4.1.2 was used. Thirteen different lab tests are represented in the model; these tests are presented in Table 5.2. Because the number of laboratory tests may vary significantly across different patients and diseases, a sample of 116 patient records were used. Each patient case included in the study was seen at

the hospital for at least three months and had at least one laboratory test performed. Variables in the graphical disease model were then used to identify a subset of laboratory tests that would be relevant for a user to examine if he/she were interested in diagnosing a patient with liver problems. On average, 9 tests were determined to be relevant out of an average of 64 tests that were performed on each patient. Using the graphical disease model to filter out data resulted in an 85.9% reduction in the number of tests that would have been displayed to the user. Historically, laboratory values are grouped and displayed together based on a predefined hierarchical organization (*e.g.*, lipid panel contains measurements of cholesterol, HDL, LDL). Traditional approaches to filter test results based on these hierarchical groups (*e.g.*, display only test results that are obtained through the lipid panel) may not provide all of the necessary tests; other lab tests outside of the selected group may be relevant for diagnosing and gauging the severity of a patient's problem. For instance, a physician diagnosing a patient with liver disease may want to see not only results from the liver panel, but also results from the lipid panel and antibody tests as well. This work thus demonstrates how the graphical disease model can be used to reorganize available lab tests into semantically related clusters, filtering out ones that are irrelevant regardless of their position in the hierarchical grouping. Additional reduction can also be achieved by filtering at more granular levels; for instance, the inclusion rules such as "include based on data value", "include based on trend", and "include based on recent activity" can be used to filter out specific data points within each test. For example, a user could request to show only abnormal values within each of the tests relevant to diagnosing hepatitis C.

5.4. Evaluation of Human Factors

Usability studies assess how well a system meets the needs of its target users. This process is especially important in medicine because systems need to complement existing clinical workflows. These studies assist with identifying potential issues in an application that might disrupt or impede the delivery of patient care. While many evaluation techniques exist, the two preliminary studies that are used to evaluate the applications described in Chapter 4 are based on the principles of formal usability testing. Users were provided with a set of tasks modeled after real-world clinical questions; they were asked to use the applications to complete these tasks. A usability survey was then administered to determine the user's overall impressions of each system. The following sections discuss the evaluation procedure and results in detail.

5.4.1. Visual Query Interface

Test subjects. The evaluation of VQI involved the participation of four users who were either radiology residents or radiologists with over five years of experience. Study participants were invited to join the evaluation by e-mail using a convenience sample. The shortcoming of this recruitment technique is that it introduces a potential sampling bias because no guarantees are made to obtain an accurate representation of the target user group. Given that the aim of this study is not to compare different user groups, convenience sampling is sufficient for the task of soliciting general impressions of the system. However, a random sample of study participants would be neces-

sary to fully determine whether users with different backgrounds (*e.g.*, cardiologist) have significantly different reactions to using VQI.

Experimental design. Each user was asked to sit in front of a simulated clinical workstation that was running a prototype version of VQI. Prior to the start of the evaluation, the user was given a brief training tutorial on how to use the system. During the tutorial, the functionality of the system was introduced (*e.g.*, main components of the interface, functionality of buttons, instructions on posing a query), and a sample task was given to the user that encouraged him/her to explore the interface and become acquainted with the visual query paradigm. Once the user indicated that he/she was ready to proceed, the user was instructed to use the interface to answer a set of 12 questions. These questions represented different clinical and research tasks that asked the user to perform prognostic queries using the underlying disease model or find similar patient cases from a large repository. The total set of questions was divided into two sets: for the first set of six randomly selected questions, the test subject was asked to use VQI with the adaptive toolbar enabled to generate the queries. After the user had completed the initial six questions, he/she was asked to disable the adaptive toolbar (*e.g.*, which forced all of the metaphors to be shown simultaneously) and answer the remaining six questions. Tasks that were asked as part of the study include:

- How many patients in the database are males diagnosed with anaplastic astrocytoma?

- Are there one or more patients in the database that are male who present with anaplastic glioma, mild edema found in the parietal lobe, and a mild oligocomponent?
- Predict the time to survival for a patient with an anaplastic astrocytoma, irregular outside edge, and moderate area of edema.

Data collection. Users were asked to write down their responses to each query and rate their answer on a Likert scale of 1 to 5, where 5 represents highest level of confidence. Following the completion of the tasks, a short usability survey modeled after the Questionnaire for User Interaction Satisfaction (QUIS) [136] was administered to each user.

Results. The primary categories measured in the usability survey are summarized in Figure 5.2. Overall, VQI scored above average on all measures with the area of interpreting results having the lowest value. Posing queries, which include functions such as selecting graphical metaphors, customizing metaphors, and formulating a visual query, received the highest positive response. Users found the interface to be intuitive and easy to learn. While three users found the adaptive toolbar to be helpful and intuitive in posing queries, one user felt that the constant addition and removal of metaphors from the toolbar was distracting and made formulating a query difficult. As evidenced in Table 5.3, all of the test participants preferred using the VQI over the traditional graph (DAG) interface of BBNs when posing imaging-related queries. While the users were split in preference when posing simple queries, most users preferred

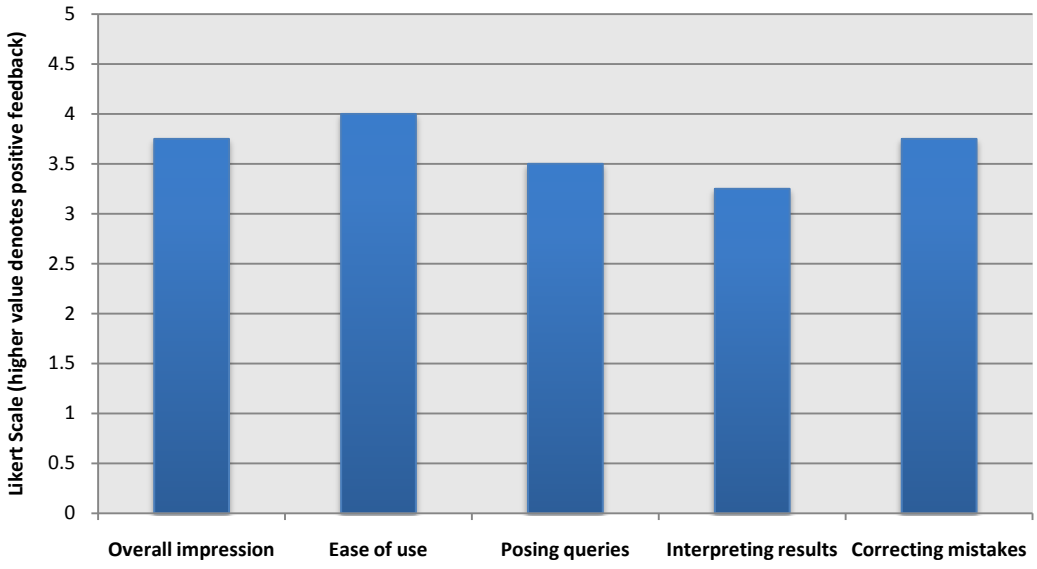


Figure 5.2: Results of the VQI user interaction questionnaire. Overall impression rates the general satisfaction of the user when using the system; it factors in the effects of the application's performance (*i.e.*, reliability, speed). Being a prototype application, the application's responsiveness and stability were suboptimal therefore lowering the user's overall impression.

Survey Question	# chosen out of 4 subjects
Select the query interface (VQI or DAG) that you would prefer to use for each of the following tasks:	
Posing queries that involve imaging features	VQI: 4 / DAG: 0
Posing simple queries that involve less than five variables	VQI: 2 / DAG: 2
Posing queries that involve more than five variables	VQI: 3 / DAG: 1

Table 5.3: Responses to a survey question asking for the user's interface preference for specific querying tasks.

VQI for queries that involved more than five variables. On average, users answered 83% of the questions correctly and reported a confidence level of 4.1. Their confidence and accuracy in answering the questions was negatively affected by difficulties with interpreting the results.

Users found several issues with the prototype user interface. Two users noted that setting the desired size of a metaphor was difficult using the visual querying paradigm. Users generally had difficulty estimating the size of metaphors drawn on the screen: particularly when sizes were mapped to ambiguous states in the model (*e.g.*, small, medium, large), users often had to draw metaphors multiple times using a trial-and-error approach before they were able to obtain the desired size. One solution would be to provide real-time information about the size of a metaphor as the user draws it. Another would be to provide UI components for modifying the size of a metaphor using an input field or drop down menu after it has been drawn. With respect to interpreting the results, users found that the visualizations used to display the most relevant cases and most probable state of a variable were difficult to read. When a case-based retrieval query is executed, a tabular display of patient cases is re-sorted based on the calculated distance between indexed cases and the query case; results are color-coded (as cases become more distant, they become proportionally darker in color) to visually represent their distance with respect to the query case. While users found identifying the highest ranking results straightforward, they had difficulty determining how well the top matches fit the query case. Users suggested that the application initially show only columns that match the variables used in the query case but provide users with the option of viewing all of the other columns. In addition, the result of the prognostic queries was difficult to interpret particularly when the probabilities of individual states were close together. A user suggested highlighting the most probable state with a different color to help users clearly differentiate it.

Limitations. The results represent a pilot evaluation that consisted of four participants who shared similar backgrounds and experience level. As VQI targets a variety of user groups, a broader study is need to determine the differences among the performance of various user groups (*e.g.*, neuro-radiologists versus thoracic radiologists). In future studies, VQI could be used alongside an existing radiology workstation; this setup may be used to measure whether the diagnostic efficiency of participants who use VQI is significantly different than those who do not have access to VQI. In addition, time required to answer each question will be recorded. Future evaluations of VQI should answer: 1) whether the adaptive toolbar reduces the time required to pose a query; 2) how easily the interface can be used by users with different levels of domain knowledge; and 3) whether the proffered answers are deemed "acceptable" to end users.

5.4.2. AdaptEHR

Test subjects. The evaluation of AdaptEHR involved the participation of six users: four informatics students who were relative novices to medicine but had an understanding of the underlying algorithms and two experienced physicians who had over five years of experience. As in the case of VQI, a convenience sample was used to recruit the study participants.

Experimental design. The objective of this study was to use the filtering and visualization capabilities of AdaptEHR to explore, relate, and present data elements in the patient record given a specific context (*e.g.*, medical problem, task). In this study, each test subject was asked to interact with a simulated clinical workstation and interpret a

patient's medical record using the AdaptEHR viewer. Users were first given a short tutorial on the functionality of the system. A sample patient case was used to demonstrate the primary features: 1) how to filter data by medical problem; 2) how to use the temporal filters; 3) how to use the semantic filters; 4) how to view the imaging and laboratory data; and 5) how to pose queries to the system. Following the guided tutorial, users were given time to familiarize themselves with the interface by experimenting with the sample patient case. Once participants were ready, they were provided with a set of 14 tasks generated from two patient cases. A subset of the tasks is provided below:

- What is the overall trend of the patient's Karnofsky performance score?
- How many documents reference the patient's problem of osteoarthritis?
- In July, 2005, there was concern whether the patient had pneumonia. Was it concluded that the patient had pneumonia?
- When was the patient's last dose of the drug temozolomide administered?

Tasks can be categorized into three types: diagnoses, prognostic predictions, and therapeutic management decisions. Participants were asked to use the AdaptEHR interface to answer each of the questions.

Data collection. Users wrote down their answers for each clinical query. Once all of the tasks were completed, they were asked to complete a short usability survey modeled after QUIS.

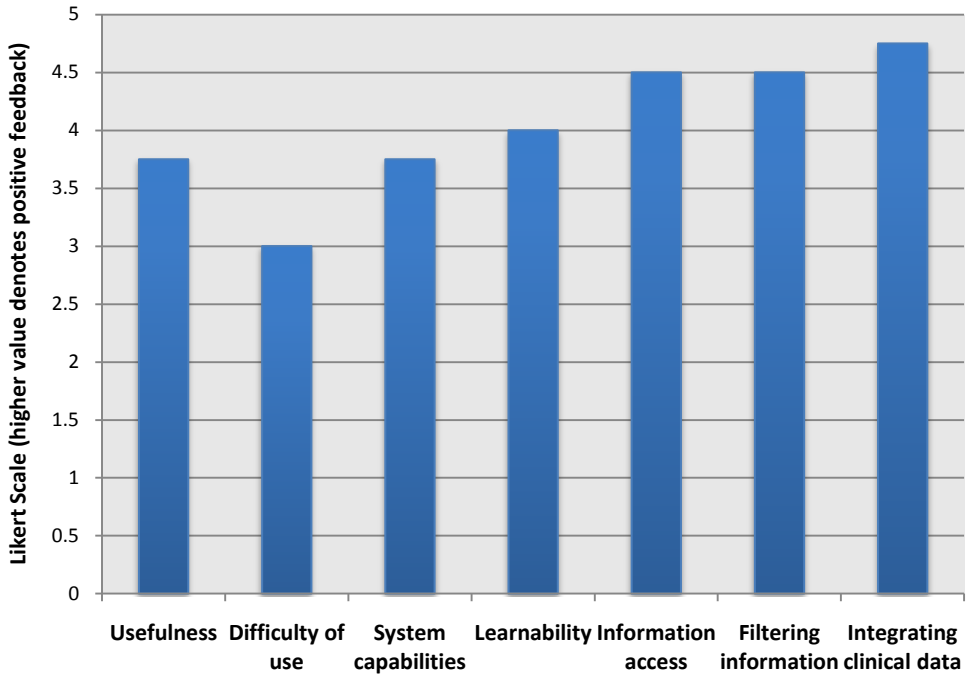


Figure 5.3: Results of the AdaptEHR user interaction questionnaire.

Results. The results of the AdaptEHR user survey were averaged and compiled into the bar graph depicted in Figure 5.3. On average, users were able to answer 86.9% of the questions correctly. They also responded with an overall positive response to the application. In particular, users appreciated the tools for filtering the information (problem list filter, semantic filter, and temporal filter), the ability to access information (*e.g.*, search and highlight functionality), and the integration of different parts of the patient record into a single display. Issues regarding AdaptEHR centered on the large amount of functionality available: several users were initially confused with how to use the filters to find relevant information. One user noted that while the UI provided multiple approaches for performing the same task, it did not provide the user

with any assistance on deciding which approach would be the most optimal. Two users also raised the issue with screen clutter on the timeline display. Despite the filtering mechanisms, some patient records had dense clusters of documents and images corresponding to periods of hospitalization or the initial diagnosis of a medical condition. In generating the display, AdaptEHR plotted the visual metaphors close together based on their timestamps, which resulted in metaphors overlapping on top of one another to the point where some documents were visually hidden underneath others. Further research into the area of graph layout mechanisms (*e.g.*, spring-embedding layout) would be helpful in laying out visual metaphors on the display in a more effective way. Additionally, users noted that the ability to define and customize user models was very limited. Advanced UI controls are needed to enhance user models with features such saving user-defined layouts and capturing information about most frequently accessed functions and data elements. In addition, techniques for learning user preferences through relevance feedback [137] need to be explored.

Limitations. Given the small sample size of users, only general conclusions may be drawn from this pilot evaluation of AdaptEHR. Similar to the evaluation of VQI, a larger pool of participants would allow users to be stratified into groups based on specialty (*e.g.*, radiologist versus primary care physician) and expertise (*e.g.*, resident versus attending physician). In addition, participants would be assigned to use either the existing health information system (*e.g.*, PCIMS, VistA) or AdaptEHR to view patient data. The expanded study would measure whether significant differences exist be-

tween current systems and AdaptEHR in the time needed to answer clinical queries and the accuracy of interpreting the information.

CHAPTER 6

Conclusion & Future Work

6. Overview

This chapter summarizes the findings and contributions that were made as a result of this dissertation. Potential avenues of research resulting from this work are also presented.

6.1. Summary & Results

This dissertation addresses the need for tools that assist users with finding, viewing, and understanding large quantities of medical data collected during routine patient care. My approach was to develop a method of context-sensitive visualization that utilizes various knowledge sources—graphical models in particular—to relate and prioritize data elements in the patient record for a given user and task. The specific contributions of this dissertation are as follows:

- *Characterization and integration of different knowledge sources to provide context.* I explored methods for extracting properties of the Bayesian belief network (BBN) by examining its variables, defined relationships, and parameters as a way to relate and prioritize elements in the patient record. I used query expansion to map variables in the model to relevant data elements in the patient record. I also described how other knowledge sources (*i.e.*, medical ontologies and user models)

can be used to supplement the information provided by the graphical model to further constrain the types of information that are displayed in the user interface.

- *A visual dictionary to translate contextual information into rules that influence how data is presented on the screen.* I created a visual dictionary to perform three tasks: 1) map data elements to available visual metaphors based on context (*e.g.*, medical condition, user); 2) change the appearance of metaphors based on properties extracted from the knowledge sources; and 3) incorporate multiple data elements into one display by following a set of composition rules that are defined by the disease model. I examined how visual cues can be used to denote the relationships between data elements by altering the size, opacity, spatial location, and layering of their visual metaphors.

I performed a review of existing knowledge sources, medical visualizations, and clinical information displays to compare my approach with existing research and to validate the originality of my contributions. I demonstrated the feasibility of my approach through two clinical applications: VQI and AdaptEHR. These applications implement the principles of context-sensitive visualization and demonstrate how adaptive user interfaces assist users with querying underlying disease models and understanding the results in the context of a patient's record. Pilot evaluations were performed to gauge the effectiveness of using graphical models to identify relevant parts of the patient record and reduce the overall amount of information presented to the user. A preliminary usability study was completed to solicit initial impressions from target users

about the two applications. Results of the pilot study were positive and supported the work's usefulness and innovation.

6.2. Future Work

While a substantial amount of work has been done to implement the algorithms and methodologies described in Chapter 3, additional work is needed to refine these approaches. My work would benefit from progress in the areas of disease modeling and information extraction. Traditional Bayesian belief networks require clinical data and disease processes—which are intrinsically temporal in nature—to be condensed and represented using a single node. Dynamic Bayesian networks (DBNs) [138], an extension of BBNs that model time-variant states, hold promise in addressing this shortcoming; however, more research is needed to create clinically useful disease models using DBNs. In addition, disease models need to become increasingly modular and extensible so that new variables may be easily incorporated into existing models. Algorithms for performing inference or characterizing medical data must be capable of handling the increased quantities of data. Improvements are also needed in the way underlying disease models are generated; models should be constructed using common data elements so that mapping between different knowledge sources become straightforward. Using a common representation for variables also has the added benefit of being able to support data from different clinical sites that follow the same standards; usage of common data elements is critical to enabling the creation of population-based disease models. With respect to other knowledge sources such as ontolo-

gies, as discussed in Section 3.4.1, one of their limitations is that they do not encode probabilistic relationships in their structures. A growing body of work addresses this issue by presenting approaches for automatically generating and annotating ontology models with values that represent uncertainty. Ding [106] describes one approach that uses a BBN as the underlying representation for capturing probabilities between concepts; the concepts, structure, and probabilities encoded in an ontology are translated into a BBN using a set of predefined rules.

In addition, fully automated methods are needed to identify and extract features from patient data because manual or semi-automated algorithms are too costly and time consuming to generate meaningful amounts of data for population-level modeling. Recent developments in screening and diagnostic techniques have resulted in improved resources for characterizing diseases and its causes. Non-invasive medical imaging, histological analysis, and gene expression profiling have shed new light on the etiology of a variety of diseases. These procedures have generated large quantities of measured data that have neither been fully explored nor translated into improved clinical care of patients [139]. In particular, genomic data has yet to be closely integrated with the patient clinical record so that the data may be viewed and understood within the context of the patient's symptoms and medical history [140]. My belief is that context-sensitive visualizations will be critical for applying the large amounts of data acquired using high throughput gene sequencing technologies towards improved care at the bedside. Current applications of information visualization in bioinformatics [141] have

primarily been designed for researchers who are experts in the area of genomic analysis. A new generation of visualizations is needed to guide lay users with finding and understanding relevant parts of these large dataset to answer questions such as: can gene expression data be used to identify which drug therapies would be most effective for a patient; or which therapies would have an adverse effect?

Finally, improvements can be made to the work presented in this dissertation. The visual dictionary is implemented as a simple flat file that defines the relationships between data elements and visualizations. More sophisticated methods for representing this information as a relational database or a probabilistic relational model should be explored. With respect to user evaluation, expanded user studies are needed. Improvements include: 1) recruiting additional participants for each user group using a random sample; 2) comparing the prototype applications with similar existing systems to determine whether any significant differences exist; and 3) performing a time-motion study to measure whether the applications significantly reduce the amount of time required to perform a task.

6.3. Concluding Remarks

The need for context-sensitive visualizations to facilitate exploration of the patient record is growing: as patient data becomes more accessible, different user groups will need a tailored visualization to navigate and understand this data. One example is the growing popularity of personal health records (PHRs). With patient data becoming more accessible by multiple parties such as patients, physicians, and administrators,

the number of users necessitates the development of unique views on the same data for each user group to meet each of their information needs. A potential extension of this research is the development of a *patient-centric medical record visualization* that educates a layperson on how to understand his/her medical record and how to act upon this information. Compared to current EMRs, which are primarily designed for use by health professionals, patient-centric interfaces need to provide clear and understandable access to patient data. Not only does the system need an intuitive interface for collecting patient information for the purpose of showing data to a primary care physician, but also, systems should use this information to help the patient understand his/her medical conditions better. For instance, rather than display lists of clinical documents, laboratory results, and medical images as would be the case in traditional EMR interfaces, a patient-centric interface would identify subsets of this data that would be of greatest pertinence to the patient: 1) generate reminders regarding when to take medications or make an appointment with a primary care provider; 2) list all of the current active medical problems and provide links to external resources (*e.g.*, Medline Plus) to learn more about them; 3) download self-reported information (*e.g.*, from sources such as Microsoft HealthVault, Google Health) or data collected by personal health devices and determine whether any trends are of concern given the patient's medical problems; and 4) highlight any laboratory test and diagnostic procedure results that are abnormal or are of concern to the patient. With the growing popularity of PHRs, patient data is becoming increasingly rich in data types and accessible; patient-centric medical records would be a significant step towards empowering the patient in

becoming more involved in their own care. The contributions of this dissertation would provide a foundation for enabling the creation of such interfaces.

References

[1] Hsu W, Bui A. Chapter 9: Disease models, Part II: Querying & Applications. In: Bui A, Taira RK, eds. *Medical Imaging Informatics*: Springer 2009:In Press.

[2] Oniśko A. Probabilistic causal models in medicine: Application to diagnosis of liver disorders. Warsaw: Polish Academy of Science; 2003.

[3] Patkar V, Fox J. Clinical guidelines and care pathways: a case study applying PROforma decision support technology to the breast cancer care pathway. Stud Health Technol Inform. 2008;139:233-42.

[4] Nease RF, JR, Owens DK. Use of influence diagrams to structure medical decisions. Med Decis Making. 1997 July 1, 1997;17(3):263-75.

[5] Wyatt JC, Wright P. Design should help use of patients' data. Lancet. 1998;352(9137):1375-8.

[6] Dey AK. Understanding and using context. Personal and Ubiquitous Computing. 2001;5(1):4-7.

[7] Bui A, Taira RK, Churchill B, Kangarloo H. Integrated visualization of problem centric urological patient records. *2001 International Conference on Mathematics and Engineering Techniques in Medicine and the Biological Sciences* 2001:413–9.

[8] Hsu W, Bui AA. A framework for visually querying a probabilistic model of tumor image features. AMIA Annu Symp Proc. 2006:354-8.

[9] Koiter JR. Visualizing inference in Bayesian networks: Delft University of Technology (Netherlands); 2006.

[10] Haddawy P, Jacobson J, Kahn CE. BANTER: a Bayesian network tutoring shell. Artificial Intelligence in Medicine. 1997;10(2):177-200.

[11] Suermondt H. Explanation in Bayesian belief networks, Ph.D. Dissertation: Stanford University; 1992.

[12] Pearl J. Probabilistic reasoning in intelligent systems: Networks of plausible inference: Morgan Kaufmann 1988.

[13] Lindberg D, Humphreys B, McCray A. The unified medical language system. Methods Inf Med. 1993;32(4):281-91.

[14] Sioutos N, Coronado S, Haber M, Hartel F, Shaiu W, Wright L. NCI Thesaurus: a semantic model integrating cancer-related clinical and molecular information. J Biomed Inform. 2007;40(1):30-43.

[15] Langlotz C. Radlex: a new method for indexing online educational materials. Radiographics. 2006;26(6):1595-7.

[16] Starren J, Johnson S. An object-oriented taxonomy of medical data presentations. Am Med Inform Assoc 2000:1-20.

[17] Shpitser I. Chapter 8: Disease models, Part I: Graphical models. In: Bui AA, Taira RK, eds. *Medical Imaging Informatics*: Springer 2009:In Press.

[18] Darwiche A. Modeling and reasoning with Bayesian networks. New York: Cambridge University Press 2009.

[19] Antal P, Meszaros T, De Moor B, Dobrowiecki T. Annotated bayesian networks: a tool to integrate textual and probabilistic medical knowledge. *Proc 14th IEEE Symposium on Computer-Based Medical Systems* 2001:177-82.

[20] Suermondt HJ, Cooper GF. An evaluation of explanations of probabilistic inference. Comput Biomed Res. 1993 Jun;26(3):242-54.

[21] Druzdzel M. Qualitative verbal explanations in Bayesian belief networks. AISB QUARTERLY. 1996:43-54.

[22] Madigan D, Mosurski K, Almond RG. Graphical explanation in belief networks. J Comput Graphical Statistics. 1996;6:160-81.

[23] Haddawy P, Jacobson J, Kahn C. BANTER: a Bayesian network tutoring shell. Artificial Intelligence in Medicine. 1997;10(2):177-200.

[24] Yap G, Tan A, Pang H. Explaining inferences in Bayesian networks. Applied Intelligence. 2008;29(3):263-78.

[25] Verduijn M, Peek N, Rosseel P, de Jonge E, de Mol B. Prognostic Bayesian networks I: rationale, learning procedure, and clinical use. J Biomed Inform. 2007;40(6):609-18.

[26] Lacave C, Dez FJ. A review of explanation methods for Bayesian networks. Knowl Eng Rev. 2002;17(2):107-27.

[27] Bodenreider O. Biomedical ontologies in action: Role in knowledge management, data integration and decision support. Yearb Med Inform. 2008:67-79.

[28] Rubin D, Dameron O, Bashir Y, Grossman D, Dev P, Musen M. Using ontologies linked with geometric models to reason about penetrating injuries. Artificial Intelligence in Medicine. 2006;37(3):167-76.

[29] Rindflesch TC, Fiszman M, Libbus B. Semantic interpretation for the biomedical research literature. In: Chen H, Fuller S, Friedman C, Hersh W, eds. *Medical informatics: Knowledge management and data mining in biomedicine*: Springer 2005:399-422.

[30] Cimino J. From data to knowledge through concept-oriented terminologies: Experience with the Medical Entities Dictionary. J Am Med Inform Assoc. 2000;7(3):288-97.

[31] Sundvall E, Nyström M, Petersson H, Åhlfeldt H. Interactive visualization and navigation of complex terminology systems, exemplified by SNOMED CT. Stud Health Technol Inform. 2006;124:851-6.

[32] Heer J, Card SK, Landay JA. Prefuse: a toolkit for interactive information visualization. *Proc of the SIGCHI conference on Human factors in computing systems*. Portland, Oregon, USA: ACM 2005:421-30.

[33] Bui AA, Hsu W. Chapter 4: Medical Data Visualization: Toward Integrated Clinical Workstations. In: Bui AA, Taira RK, eds. *Medical Imaging Informatics* 2009:In Press.

[34] Rich E. User modeling via stereotypes. Cognitive Science. 1979;3(4):329-54.

[35] Limbourg Q, Vanderdonckt J. Comparing task models for user interface design. In: Diaper D, Stanton N, eds. *The Handbook of Task Analysis for Human-Computer Interaction* 2003:135-54.

[36] Lewis C, Rieman J. Task-centered user interface design: A practical introduction. University of Colorado, Boulder 1993.

[37] Field M, Lohr K. Guidelines for clinical practice: from development to use: Natl Academy Pr 1992.

[38] Grimshaw J, Russell I. Effect of clinical guidelines on medical practice: a systematic review of rigorous evaluations. Lancet. 1993;342(8883):1317-22.

[39] Shahar Y, Young O, Shalom E, Galperin M, Mayaffit A, Moskovitch R, et al. A framework for a distributed, hybrid, multiple-ontology clinical-guideline library, and automated guideline-support tools. J Biomed Inform. 2004;37(5):325-44.

[40] Shortliffe EH, Scott AC, Bischoff MB, Campbell AB, van Melle W, Jacobs CD. ONCOCIN: An expert system for oncology protocol management. Proc 17th Int Joint Conf on Artificial Intelligence. 1981:876–81.

[41] Fox J, Johns N, Lyons C, Rahmanzadeh A, Thomson R, Wilson P. PROforma: a general technology for clinical decision support systems. Computer Methods and Programs in Biomedicine. 1997;54(1-2):59-67.

[42] Shiffman RN, Karras BT, Agrawal A, Chen R, Marenco L, Nath S. GEM: A Proposal for a More Comprehensive Guideline Document Model Using XML. J Am Med Inform Assoc. 2000 September 1, 2000;7(5):488-98.

[43] Peleg M, Boxwala A, Ogunyemi O, Zeng Q, Tu S, Lacson R, et al. GLIF3: the evolution of a guideline representation format. 2000: American Medical Informatics Association; 2000. p. 645.

[44] Sutton DR, Fox J. The Syntax and Semantics of the PROforma Guideline Modeling Language. J Am Med Inform Assoc. 2003 September 1, 2003;10(5):433-43.

[45] Shneiderman B, Card S, Mackinlay J. Readings in information visualization: using vision to think: Morgan Kaufmann 1999.

[46] Eisen MB, Spellman PT, Brown PO, Botstein D. Cluster analysis and display of genome-wide expression patterns. Proc National Academy of Sciences. 1998 December 8, 1998;95(25):14863-8.

[47] Hoeke J, Gelsema E, Wulkan R, Leijnse B. Graphical non-linear representation of multi-dimensional laboratory measurements in their clinical context. Methods Inf Med. 1991;30(2):138-44.

[48] Sheth N, Cai Q. Visualizing MeSH dataset using radial tree layout. Bloomington: Indiana University; 2003.

[49] Bodenreider O. Experiences in visualizing and navigating biomedical ontologies and knowledge bases. *Proc Intelligent Systems for Molecular Biology (ISMB) SIG Meeting (Bio-ontologies)* 2002.

[50] Payne PR, Starren JB. Quantifying visual similarity in clinical iconic graphics. J Am Med Inform Assoc. 2005 May-Jun;12(3):338-45.

[51] Cho H, Ishida T, Yamashita N, Inaba R, Mori Y, Koda T. Culturally-situated pictogram retrieval. In: Ishida T, Fussell SR, Vossen PTJM, eds. *Connecting the Universal to the Specific: Towards the Global Grid (IWIC 2007)* 2007:211-35.

[52] Ramer L, Richardson JL, Cohen MZ, Bedney C, Danley KL, Judge EA. Multimeasure pain assessment in an ethnically diverse group of patients with cancer. J Transcult Nurs. 1999;10(2):94-101.

[53] Knapp P, Raynor DK, Jebar AH, Price SJ. Interpretation of medication pictograms by adults in the UK. Ann Pharmacotherapy. 2005;39(7):1227-33.

[54] Hearst M. TileBars: Visualization of term distribution information in full text information access. *Proc SIGCHI Conf Human Factors in Computing Systems*. Denver, CO, United States: ACM Press/Addison-Wesley Publishing Co. 1995.

[55] Lin X. Visualization for the document space. *Proc IEEE Conf Visualization '92* 1992:274-81.

[56] Dix A, Ellis G. Starting simple: Adding value to static visualisation through simple interaction. *Proc Working Conf Adv Visual Interfaces* 1998:124-34.

[57] Tweedie L. Characterizing interactive externalizations. *Proc SIGCHI Conf Human Factors in Computing Systems* 1997:375-82.

[58] Ahlberg C, Williamson C, Shneiderman B. Dynamic queries for information exploration: An implementation and evaluation. *Proc ACM SIGCHI Conf Human Factors in Computer Systems* 1992:619-26.

[59] Shneiderman B. Direct manipulation: A step beyond programming languages 1981.

[60] Hansen MD. An analysis of the diagrammatic visual data querying domain [dissertation]. Santa Cruz: University of California, Santa Cruz; 2005.

[61] Hansen MD. A survey of systems in the diagrammatic visual data querying domain: University of California, Santa Cruz; 2005.

[62] Eick SG. Visualizing multi-dimensional data. ACM SIGGRAPH Computer Graphics. 2000;34(1):61-7.

[63] Ahlberg C. Spotfire: an information exploration environment. ACM SIGMOD Record. 1996;25(4):25-9.

[64] Ogunyemi O. Methods for reasoning from geometry about anatomic structures injured by penetrating trauma. J Biomed Inform. 2006;39(4):389-400.

[65] Chang NS, Fu KS. Query-by-pictorial-example. Proc 3rd Int Conf on Computer Software and Applications. 1979:325-30.

[66] Joseph T, Cardenas AF. PICQUERY: a high level query language for pictorial database management. IEEE Transactions on Software Engineering. 1988;14(5):630-8.

[67] Dionisio JDN, Cardenas AF. MQuery: A visual query language for multimedia, timeline and simulation data. Journal of Visual Languages and Computing. 1996;7:377-401.

[68] Del Bimbo A, Pala P. Image retrieval by elastic matching of user sketches. *Image Analysis and Processing* 1995:185-90.

[69] Cruz IF. DOODLE: a visual language for object-oriented databases. *Proc 1992 ACM SIGMOD international conference on Management of data*. San Diego, California, United States: ACM 1992:71-80.

[70] Egenhofer MJ. Query processing in spatial-query-by-sketch. Journal of Visual Languages and Computing. 1997;8(4):403-24.

[71] Abate AF, Nappi M, Tortora G, Tucci M. Assisted browsing in a diagnostic image database. *Proc of the Workshop on Advanced visual interfaces* 1996:223-32.

[72] Sasso G, Marsiglia HR, Pigatto F, Basilicata A, Gargiulo M, Abate AF, et al. A visual query-by-example image database for chest CT images: Potential role as a decision and educational support tool for radiologists. J Digital Imaging. 2005;18(1):78-84.

[73] Shortliffe EH. The evolution of electronic medical records. Acad Med. 1999;74(4):414-9.

[74] Kilsdonk A, Frandji B, van der Werff A. The NUCLEUS integrated electronic patient dossier breakthrough and concepts of an open solution. International Journal of Bio-Medical Computing. 1996;42(1-2):79-89.

[75] de Estrada W, Murphy S, Barnett G. Puya: a method of attracting attention to relevant physical findings. AMIA Annu Symp Proc; 1997; 1997. p. 509-13.

[76] Zeng Q, Cimino JJ. A knowledge-based, concept-oriented view generation system for clinical data. J Biomed Inform. 2001;34(2):112-28.

[77] Weed L. Medical records that guide and teach. N Engl J Med. 1968;278(11):593-600.

[78] Cousins S, Kahn M. The visual display of temporal information. Artificial Intelligence in Medicine. 1991;3(6):341-57.

[79] Plaisant C, Mushlin R, Snyder A, Li J, Heller D, Shneiderman B. LifeLines: using visualization to enhance navigation and analysis of patient records. 1998: American Medical Informatics Association; 1998. p. 76.

[80] Wang TD, Plaisant C, Quinn AJ, Stanchak R, Murphy S, Shneiderman B. Aligning temporal data by sentinel events: discovering patterns in electronic health records. *Proc of the 26th Annual SIGCHI Conf on Human Factors in Computing Systems*. Florence, Italy: ACM 2008:457-66.

[81] Bui AAT, Aberle DR, Kangarloo H. TimeLine: Visualizing integrated patient records. IEEE Trans Inf Technol Biomed. 2007;11(4):462-73.

[82] Boonn W, Langlotz C. Radiologist use of and perceived need for patient data access. J Digit Imaging. 2008:In press.

[83] Ratib O, Swiernik M, McCoy JM. From PACS to integrated EMR. Computerized Medical Imaging and Graphics. 2003;27(2-3):207-15.

[84] Weed LL. Medical records that guide and teach. N Engl J Med. 1968 Mar 21;278(12):652-7 concl.

[85] Tory M, Moller T. Human factors in visualization research. Visualization and Computer Graphics, IEEE Transactions on. 2004;10(1):72-84.

[86] Langley P, Hirsh H. User modeling in adaptive interfaces: Autonomous Agents 2000.

[87] Brusilovsky P. Adaptive Hypermedia. User Modeling and User-Adapted Interaction. 2001;11(1):87-110.

[88] Debevc M, Meyer B, Donlagic D, Svecko R. Design and evaluation of an adaptive icon toolbar. User Modeling and User-Adapted Interaction. 1996;6(1):1-21.

[89] Horvitz E, Breese J, Heckerman D, Hovel D, Rommelse K. The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users. *Proc 14th Conf on Uncertainty in Artificial Intelligence* 1998:256–65.

[90] Francisco-Revilla L, Shipman III F. Adaptive medical information delivery combining user, task and situation models. *Proc of the 5th Int Conf on Intelligent User Interfaces* 2000:94-7.

[91] Cimino JJ, Li J, Bakken S, Patel VL. Theoretical, empirical and practical approaches to resolving the unmet information needs of clinical information system users. *Proc AMIA Symp* 2002:4.

[92] Nygren E, Wyatt J, Wright P. Helping clinicians to find data and avoid delays. The Lancet. 1998;352:1462-66.

[93] Qiu Y, Frei H. Concept based query expansion. 1993: ACM New York, NY, USA; 1993. p. 160-9.

[94] Ide N, Loane R, Demner-Fushman D. Essie: a concept-based search engine for structured biomedical text. J Am Med Inform Assoc. 2007;14(3):253-63.

[95] Liu Z, Chu W. Knowledge-based query expansion to support scenario-specific retrieval of medical free text. Information Retrieval. 2007;10(2):173-202.

[96] Zeng QT, Crowell J, Plovnick RM, Kim E, Ngo L, Dibble E. Assisting consumer health information retrieval with query recommendations. J Am Med Inform Assoc. 2006;13(1):80-90.

[97] Zeng Q, Cimino JJ, Zou KH. Providing concept-oriented views for clinical data using a knowledge-based system: An evaluation. J Am Med Inform Assoc. 2002;9(3):294.

[98] Newman MEJ. The structure and function of complex networks. SIAM Review. 2003;45(2):167-256.

[99] Dijkstra E. A note on two problems in connexion with graphs. numerische Mathematik. 1959;1(1):269-71.

[100] Castillo E, Gutierrez JM, Hadi AS. Sensitivity analysis in discrete Bayesian networks. Systems, Man and Cybernetics, Part A, IEEE Transactions on. 1997;27(4):412-23.

[101] Laskey KB. Sensitivity analysis for probability assessments in Bayesian networks. Systems, Man and Cybernetics, IEEE Transactions on. 1995;25(6):901-9.

[102] Kullback S, Leibler RA. On information and sufficiency. The Annals of Mathematical Statistics. 1951;22(1):79-86.

[103] Jensen FV. Bayesian networks and decision graphs. New York: Springer 2001.

[104] Haddawy P, Jacobson J, Kahn CE. BANTER: A Bayesian network tutoring shell. Artif Intell Med. 1997;10(2):177-200.

[105] Bui AA, Taira RK, Churchill B, Kangarloo H. Integrated visualization of problemcentric urologic patient records. Annals of the New York Academy of Sciences. 2002;980:267-77.

[106] Ding Z, Peng Y. A probabilistic extension to ontology language OWL. *Proc of the 37th Annual Hawaii Int Conf on System Sciences*: IEEE Computer Society 2004:1-10.

[107] Lin Y, Li W, Chen K, Liu Y. A document clustering and ranking system for exploring MEDLINE citations. J Am Med Inform Assoc. 2007;14(5):651-61.

[108] Hsu W, Bui AA, Taira RK, Kangarloo H. Chapter II: Integrating imaging and clinical data for decision support. In: Exarchos T, A P, DI F, eds. *Handbook of Research on Advanced Techniques in Diagnostic Imaging and Biomedical Applications*. Hershey: IGI Global 2009.

[109] Ali K, Hartmann K, Fuchs G, Schumann H. Adaptive layout for interactive documents. *Proc of the 9th Int Symp on Smart Graphics*. Rennes, France: Springer-Verlag 2008:247-54.

[110] Senay H, Ignatius E. A knowledge-based system for visualization design. IEEE Computer Graphics and Applications. 1994;14(6):36-47.

[111] Mackinlay J. Automating the design of graphical presentations of relational information. ACM Trans Graph. 1986;5(2):110-41.

[112] Shneiderman B. The eyes have it: A task by data type taxonomy for information visualizations. *Proc IEEE Symp Visual Languages* 1996:336-43.

[113] Marcus A. Metaphor design for user interfaces. Conference on Human Factors in Computing Systems. 1998:129-30.

[114] Chen P, Aldape K, Wiencke JK, Kelsey KT, Miike R, Davis RL, et al. Ethnicity delineates different genetic pathways in malignant glioma. Cancer Res. 2001 May 1, 2001;61(10):3949-54.

[115] Siau K, Chan H, Wei K. Effects of query complexity and learning on novice user query performance with conceptual and logical database interfaces. IEEE Transactions on Systems, Man and Cybernetics, Part A. 2004;34(2):276-81.

[116] Hsu W, Bui A. A framework for visually querying a probabilistic model of tumor image features. AMIA Annu Symp Proc; 2006; 2006. p. 354-8.

[117] Rex D, Ma J, Toga A. The LONI pipeline processing environment. Neuroimage. 2003;19(3):1033-48.

[118] Darwiche A. SamIam. University of California, Los Angeles, Automated Reasoning Group 2009.

[119] Kullback S, Leibler RA. On information and sufficiency. Annals Mathematical Statistics. 1951;22:79-86.

[120] Tange HJ, Hasman A, de Vries Robbe PF, Schouten HC. Medical narratives in electronic medical records. Int J Med Inform. 1997;46(1):7-29.

[121] Bossen C. Evaluation of a computerized problem-oriented medical record in a hospital department: Does it support daily clinical practice? Int J Med Inform. 2007;76(8):592-600.

[122] Bui AA. A multimedia data model with generalized stream constructs: University of California, Los Angeles; 2000.

[123] Friedman C, Shagina L, Lussier Y, Hripcsak G. Automated encoding of clinical documents based on natural language processing. J Am Med Inform Assoc. 2004;11(5):392-402.

[124] Meystre S, Haug P. Automation of a problem list using natural language processing. BMC Medical Informatics and Decision Making. 2005;5(1):30.

[125] Bashyam V, Hsu W, Watt E, Bui AAT, Kangarloo H, Taira RK. Informatics in radiology: problem-centric organization and visualization of patient imaging and clinical data. Radiographics. 2009 March 1, 2009;29(2):331-43.

[126] Bodenreider O, McCray A. Exploring semantic groups through visual approaches. J Biomed Inform. 2003;36(6):414-32.

[127] Taira R, Bashyam V, Kangarloo H. A field theoretical approach to medical natural language processing. IEEE Trans Inf Technol Biomed. 2007;11(4):364-75.

[128] Bui AAT, Morioka C, Dionisio JDN, Johnson DB, Sinha U, Ardekani S, et al. openSourcePACS: An Extensible infrastructure for medical image management. IEEE Trans Inf Technol Biomed. 2007;11(1):94-109.

[129] Druzdzel MJ. SMILE: Structural Modeling, Inference, and Learning Engine and GeNIe: a development environment for graphical decision-theoretic models. *Proc of the 16th National Conf on Artificial intelligence*. Orlando, Florida, United States 1999:902-3.

[130] Burkle T, Ammenwerth E, Prokosch H, Dudeck J. Evaluation of clinical information systems. What can be evaluated and what cannot? Journal of Evaluation in Clinical Practice. 2001;7(4):373.

[131] Nadkarni P, Chen R, Brandt C. UMLS concept indexing for production databases: A feasibility study. J Am Med Inform Assoc. 2001;8(1):80-91.

[132] Aronson A. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. AMIA Annu Symp Proc; 2001; 2001. p. 17-21.

[133] Winget MD, Baron JA, Spitz MR, Brenner DE, Warzel D, Kincaid H, et al. Development of common data elements: the experience of and recommendations from the early detection research network. Int J Med Inform. 2003;70(1):41-8.

[134] Stevenson M. Word sense disambiguation: the case for combinations of knowledge sources. Computational Linguistics.30(2).

[135] Matykiewicz P, Duch W, Pestian J. Nonambiguous concept mapping in medical domain. *Artificial Intelligence and Soft Computing – ICAISC 2006* 2006:941-50.

[136] Chin JP, Diehl VA, Norman KL. Development of an instrument measuring user satisfaction of the human-computer interface. *Proc of the SIGCHI conference on Human factors in computing systems*. Washington, D.C., United States: ACM 1988:213-8.

[137] Sun Z, Li B, Wang Q, Feng G. Dynamic user modeling for sketch-based user interface. *Technologies for E-Learning and Digital Entertainment* 2006:1268-73.

[138] Murphy KP. Dynamic bayesian networks: representation, inference and learning: University of California, Berkeley; 2002.

[139] West M, Ginsburg G, Huang A, Nevins J. Embracing the complexity of genomic data for personalized medicine. Genome Research. 2006;16(5):559.

[140] Hoffman MA. The genome-enabled electronic medical record. Journal of Biomedical Informatics. 2007;40(1):44-6.

[141] Tao Y, Friedman C, Lussier Y. Visualizing information across multidimensional post-genomic structured and textual databases. Bioinformatics. 2005;21(8):1659-67.