2011

Design Space Decomposition Using Concept Maps

David E. Robbins
University of Alabama at Birmingham

DESIGN SPACE DECOMPOSITION USING CONCEPT MAPS

by

DAVID E. ROBBINS

MURAT M. TANIK, COMMITTEE CHAIR
BARRETT R. BRYANT
MURAT N. TANJU

A THESIS

Submitted to the graduate faculty of the University of Alabama at Birmingham
in partial fulfillment of the requirements for the degree of
Master of Science

BIRMINGHAM, ALABAMA

2011

DESIGN SPACE DECOMPOSITION USING CONCEPT MAPS

DAVID E. ROBBINS

ELECTRICAL ENGINEERING

ABSTRACT

Decomposition of engineering design spaces has been performed primarily through statistical and system identification techniques. However, no approach has yet attempted to capture the semantics of a design through decomposition. This thesis presents graphical techniques for capturing the semantics of a design by developing an information architecture through design space decomposition using concept maps. In particular, concept maps are used to begin identifying the top layers of the information architecture of a problem, which has the dual effects of performing an initial decomposition of the design space and refining the parameters that will define more detailed design spaces for further decomposition using traditional techniques. By demonstrating these techniques in two case studies, the techniques are shown to capture the semantics of a design in a computable format.

Keywords: Design Space Decomposition, Concept Maps, Semantics

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Herbert Simon initiated the modern study of design, in part, by proposing a method for hierarchically decomposing the parameter space associated with a complex system according to the strengths of relationships between parameters [1]. Conant formalized this method using the concept of entropy [2]. Numerous other statistical decomposition techniques were developed and eventually linked in a unified framework [3]. These decomposition techniques, however, do not address the semantic nature of the system in question.

Furthermore, as the problems engineers are called to solve become increasingly complex, new tools are required to maintain intellectual control over both the breadth and the depth of the solutions we create. Intellectual control not only provides enhanced quality by reducing defects in the work of designers and engineers, but also improves the productivity of those engineers by enabling them to know what to design at any given moment [4,5].

Additionally, the practice of systems engineering continues to expand, and the work of engineers is continually moving to higher levels of abstraction. The move to component-based-engineering [6], where engineers wire together systems-of-systems from preexisting elements, requires a new way of approaching the design space decomposition problem.

Information architecture represents the fundamental concept encompassing each of these issues of modern engineering. The information architecture of a system answers the fundamental, “what does it do?” question for both the overall system and the subsystems it contains. Analyzing the organization and interfaces of the set of processes that comprise the system, and capturing that analysis in a semantic format, answers this question.

This thesis presents graphical techniques for semantically developing an information architecture through design space decomposition using concept maps. In particular, concept maps are used to identify the top layers of the information architecture of a problem, which has the dual effects of performing an initial decomposition of the design space and refining the parameters that will define more detailed design spaces for further decomposition using traditional techniques.

Project Overview

Creating techniques for concept map based design space decomposition comprised three overall activities:

1. Development and description of the proposed techniques,
2. Demonstration of the techniques in two case studies,
3. Evaluation of the techniques as applied to the case studies.

The developed technique for design space decomposition using concept maps consists of three phases. During the first phase, the top-level information architecture is identified, named, captured as a concept in a concept map, and given a brief textual description. The second phase consists of identifying a set of subsystems composing the top-level information architecture, and capturing their relationships to each other and to the top-level information architecture in the concept map. Finally, during the third phase, subsystem elaboration is performed on each of the identified subsystems, descending hierarchically until the desired level of detail is reached.

Two case studies were developed to demonstrate the proposed techniques: an n-queens based combinatorics education website and a clinical decision support system. Evaluation of these case studies showed design space decomposition successfully captured the semantic nature of the systems under design. The computerized tools used to generate the concept maps, however, lack effective collaboration models.

Thesis Outline

First, a discussion of the fundamental background supporting design spaces, concept maps, information architecture, and decomposition is provided. Presentation of the design space decomposition with concept maps techniques follows. Next, the techniques are demonstrated in two case studies. Finally, following a discussion of the utility of the techniques proposed and tools used, concluding remarks are given.

BACKGROUND

In *The Sciences of the Artificial*, Herbert Simon initiated a scientific study of the design process and the artifacts it creates. Design, according to Simon, fundamentally consists of determining a process for improving upon existing situations (pp. 111 of [1]). For most engineering problems, the process of design will include an optimization of a set of variables, within a set of constraints, to minimize the value of a cost function according to a set of environmental parameters. If the variables are taken to define a multi-dimensional space, the task of finding the optimum within that space represents the definition of design space exploration.

For a significant portion of design problems, the number and range of the variables create a design space too large to practically explore using mathematical optimization techniques. Thus, techniques are required to reduce the size of the design space and to determine satisfactory (rather than optimal) solutions. Limiting the design space, however, must be done carefully to avoid information loss and elimination of valid design alternatives [7]. Decomposition, or partitioning, of the design space represents a key strategy for reducing the overall size of a design space under consideration.

Simon also proposed an intellectual framework for hierarchical decomposition of complex spaces by grouping sets of variables according to the strength of their relationships (pp. 204 of [1]). This proposed method was mathematically formalized for variable systems through the use of entropy [2]. Other statistical methods for decomposing variable systems have been proposed and unified in a comprehensive framework [3].

Since many of the design problems engineers face today do not have well known parameter sets, current decomposition techniques cannot be applied. Additionally, even when all parameters are known, current decomposition techniques fail to capture the semantics of the system under consideration. Both of these shortcomings are here remedied through the application of concept maps to design space decomposition.

System, Process, Design, and Metrics

Knowledge of the concepts of system, process, design and metrics, and their interactions, represents a fundamental requirement for deep understanding of the world, especially from the perspective of scientists and engineers.

Process

Processes represent change as a series of steps or activities, each potentially containing sub-processes. In the context of engineering design, a process is an activity that serves as part of a solution to a problem (i.e., a situation requiring improvement). A process may contain only a single activity, and the time scale of the activity may be at any order of magnitude.

Systems

Systems are collections of processes working together towards a single goal. Systems are described by their overall function (i.e., the shared purpose) and their architecture. Architecture, in turn, comprises the set of subsystems, their organization or structure in relation to each other, and descriptions of their interfaces and the information flow between them.

To adopt this view of systems, engineers must recognize that all physical objects can be viewed as processes. A hammer, for instance, could be considered as a collection of processes for transmitting and multiplying forces. Taking this abstract view of physical objects provides the added benefit of "outside the box" thinking.

Design

A design (the noun form of the word) is a description of a system for humans. Constructing, or realizing, a design creates a description of that system for the entities that will execute the processes. Designs provide an image of a system in a single instant, or a collection of instances, of time.

Metrics

Metrics are parameters used to describe systems, processes, and designs. Classical metrics, such as weight, length, opacity, and temperature are included. Metrics determine how well or how poorly a system fulfills its design.

They may also be used as discovery tools for describing natural systems. Metrics and measurements are inherently subjective; their outcome always depends on the perspective of the observer taking the measurement.

Design Space Decomposition

Design space decomposition refers to the partitioning of the total space of all possible design solutions to an engineering problem. The dimensions of this space are the various parameters that may be used to describe the system-of-systems under design. Decomposing the design space into smaller spaces simply limits the values certain parameters may take for each sub-space.

As in any partitioning, there are multiple interdependent ways to decompose a design space. That is, several overlapping sets of system-of-systems elements may be generated, with each set representing a unique way to meet the functional and non-functional requirements.

As an analogy, consider the different ways a human and a dog would partition the energy in a room [8]. A human, relying on sight, would observe object boundaries based on the photonic energy reaching their eye. Thus, they would observe tables, chairs and other objects by their optical properties. A dog, however, relies on the sense of smell. With this sense, they would partition the room differently: the area near the person, the area near the trash, a spot where a dog sat last month, etc. Both the human and the dog are observing the same room, but they partition it differently, based on their method of decomposition.

The concept of decomposing a design space to find satisfactory and optimal solutions found numerous applications in the artificial intelligence and space exploration communities [9-12]. Initial emphasis was placed on parameter identification and the adaptation of optimization algorithms [11]. The use of traditional numerical optimization [11] and genetic algorithms [13,14] were both popular.

Searching a design space for a single optimal design is also referred to as design space exploration [15,16]. Although the terms are used almost interchangeably in the literature, we use design space decomposition to refer to a partitioning of the design space into regions either as components or as areas of high and low performance. We consider design space exploration as the pursuit of a single optimal design, an exercise that follows after the partitioning of the design space uses the methods we propose.

The study of design space decomposition includes breaking a problem into sub-problems to limit the number of parameters being optimized at a given time [9,10,12]. Chandrasekaran in particular points out the necessity of strategies to shrink the total design space before the use of computerized optimization [9]. This shrinking of the design space has the dual effect of lessening the computational load required to decompose the space while simultaneously increasing the optimality of the discovered solutions. However, care must be taken to avoid information loss in this process [7].

Our methods focus on the second emphasis, or the decomposition of a problem into sub-problems. In our technique, concept maps are used by the designer to perform this initial space-shrinking by breaking the system to be designed into a hierarchical, layered, or networked system-of-systems.

Chandrasekaran also points out that the deliberative process of searching for optimal parameter-of-interest values does not exclude the often intuitive nature of design [9]. Rather, the deliberative processes validate the intuitive design decisions. Our technique seeks to more directly link the intuitive and deliberative elements of design by placing them together in the common graphical framework of concept maps. This linkage is accomplished by capturing the semantics of the design in a format that is both readable by humans and immediately computable as a semantic object.

Parmee and Beck’s goal, with their strategy of design space decomposition using genetic algorithms, is not to find an optimal system-of-systems architecture, but rather to find regions of high performance in the design space [13]. Once such regions are discovered, traditional optimization techniques and qualitative analysis can be used to select the true optimal design from within that region. Our concept map based technique provides the framework for moving between these types of analysis easily, as well as making the qualitative relationships between separate elements of the system-of-systems more apparent to the designer.

The parameters defining a design space may consist of a variety of types: numerical values [11], system structure descriptions [14], and abstract concepts such as usage modality [17]. While our techniques provide a framework for the inclusion of each of these parameter types in the decompositions they develop, our emphasis is placed on the system structure, or architectural, parameters. This structural emphasis allows our technique to more readily manage the complexity of modern systems [18-21].

Finally, recent research into the use of interactive evolutionary computing for design space decomposition has underscored the benefit of designer intervention [15,22]. Our system will build on this concept, emphasizing the direct decomposition of the design space by the designer using concept map oriented tools.

Concept Maps

Although concept maps have been around for over thirty years, the overwhelming majority of the research emphasizes their use in education [23-26]. Recently, however, their use for research and design tasks has been explored, with promising results [27,28].

Concept maps are graphical tools that represent knowledge in the form of a directed graph with concepts as vertices and the relationships between those concepts shown as edges. This differentiates them from mind-maps, which do not generally include linking phrases, and place less emphasis on the forming of complete thoughts using concepts connected by linking phrases [23].

Since concept maps represent data in the form of dyadic predicates, their content is directly semantic. That is, all semantic statements can be reduced into a collection of dyadic predicates. This represents the classic “triple” of the semantic web.

Current computerized tools for concept mapping allow for sharing of concept maps between multiple users (e.g., customers and designers), and include the ability to link other types of data to individual concepts [29]. Additionally, these tools provide the ability to export a concept map in eXtensible Markup Language (XML) format, which provides the ability to perform further computations (e.g., design space decompositions using traditional techniques) on the information provided in the concepts [27,30]. Of particular interest is the ability to export the semantics of a concept map in a semantic web format, such a Resource Description Framework (RDF) or Web Ontology Language (OWL).

Current concept mapping software primarily provides for educational applications, following a similar pattern to the research in this area. Repurposing this software for design work has been tentatively explored [27,28]. Most importantly, concept maps capture the semantics of a design in a computable format [31].

Information Architecture

Information architecture refers to the information content of a system, and emphasizes the structure of that information. Recalling that a system is merely a collection of processes working together towards a single goal, the information architecture of a system describes the unifying purpose the processes of the system work together to attain. An information architecture answers the question "what does it do?" The answer provided by information architecture represents the fundamental semantic nature of the system.

Typically, an information architecture requires a layered set of subsystems, with each layer representing abstractions of the layer beneath [18]. Each process within a system can be viewed as a system in its own right. Thus, each component process may have its own information architecture, which is in turn an element of the overall information architecture of the system. These component information architectures each answer the same semantic question for their respective process: "what does it do?"

An information architecture may include descriptions of uses, users, physical properties, and constraints. How, then, does it differ from a traditional requirements specification? An information architecture emphasizes an awareness of systems, processes, designs, and metrics. Information architectures also generally represent a higher level of abstraction than detailed requirements specifications. Finally, an information architecture focuses on the semantics of a system and its processes.

The structure of an information architecture closely resembles that of traditional systems architecture. Three key systems architectures exist: hierarchies, networks, and layers [19]. Political, engineering, social, economic, and business systems all rely on these architectures to provide the structural frameworks on which they are built. In many cases, these architectures are not found in a pure state; system architects generally create hybridizations of the systems in order to balance the strengths of one against the weakness of the others.

While each of the three major architectures provides it's own strengths, the layered architecture appears to be the strongest [19]. Combining the levels of abstraction available to a hierarchy with the flexibility of a network, layered systems represent some of our most successful and long-lived creations. The Transmission Control Protocol/Internet Protocol (TCP/IP) combination is a classic example of the success of layered systems. Each of the five layers communicates only with the layers immediately above and below. This approach has allowed adaptation to new technologies, such as wireless networking, by simply swapping out the component at a given layer (in this case, the physical layer). Standardized communication between layers ensures that new technologies will be readily integrated when they are created.

CONCEPT MAPS FOR DESIGN SPACE DECOMPOSITION

Design space decomposition using concept maps comprises three key phases: top-level information architecture identification, subsystem elaboration, and layered detailing. The phases descend hierarchically through layers of abstraction, and may be iterated until an arbitrary level of detail is reached.

Top Level Information Architecture Identification

Decomposing the design space with concept maps begins with the identification of the overall goal of the system under design. This overall goal will tie the processes of the system into a coherent whole. For many systems, a simple statement of the system's purpose in the form of a use case scenario is sufficient.

The top level information architecture is then captured in a concept map, initially, as a single concept naming the system. A description of the systems overall functionality is then added to this concept. This description provides the basis for the systems elaboration phase.

System Elaboration

After identifying the top level information architecture, the subsystems that combine to generate the desired overall behavior are identified and likewise named. Several strategies can be used to determine these subsystems.

From the description added to the single concept naming the system, several orthogonal processes may be identified. As these processes are identified, they are likewise named in concepts and linked to the top-level concept with appropriate linking phrases.

Another strategy consists of simply making concept blocks for as many processes making up the desired overall functionality. The goal of this initial concept generation is to be as prolific as possible, naming as many subsystems as can be identified. The subsystems identified may be at any level of abstraction; the outcome of this exercise is improved with increased detail. These subsystems are then grouped into overall areas of functionality, with each area of functionality representing an abstraction of the identified sub-processes. The description for this process is then generated from the subsystems it comprises.

Once two or more such orthogonal subsystems are identified, relationships between the subsystems may also be identified and captured with linking phrases. These subsystem relationships form the interfaces that allow the subsystems to work together to achieve the overall goal of the system. In general, these relationships also determine the nature of the information architecture at that level of abstraction (layered, networked, or hierarchical).

Layered Detailing

The system elaboration process may be continued for each sub process identified, continuing through layers of abstraction until the desired level of detail is reached. Once the most detailed subsystems are identified, the parameters of each of these systems combine to make up a partitioned design space. The layered detailing phase also includes attachment of supplementary materials, such as documents, standards, images, or URL's, to each conceptual block.

As a system is identified and its subsystems are elaborated, a designer descends through levels of abstraction to ever more detail. The language used to describe the relationships between processes at various levels of abstraction in some ways resembles the language used to describe a hierarchical or tree-like information architecture. In this case, however, the resemblance is purely syntactic; the information architecture of a system at a given layer of abstraction is not generally confined to a network, layered, or hierarchical approach.

The parameter sets of each of the most detailed subsystems in the information architecture may overlap. For instance, if a data-storage parameter is required by several subsystems, the value of that parameter (in this case representing a method of data storage, e.g., relational database or XML files) must be considered individually for each subsystem.

However, in some cases an overlapping parameter may point to the need for an additional shared subsystem. When a shared subsystem is used, the value of the overlapping parameter is shared by all applicable subsystems. This may represent the optimal design by sharing the costs associated with that parameter among several subsystems. Conversely, the overall design may suffer if the optimal shared parameter value for some subsystems is suboptimal for others.

Determining whether overlapping parameters are best shared or considered individually points to the need for design space exploration at multiple levels of abstraction.

CASE STUDIES

N-Queens Based Combinatorics Education Website

Introduction

The n-queens.com website has as it's general goals combinatorics education using the N-Queens problem; teaching the concepts of process, system, design, and metric; and serving as the definitive resource for n-queens material on the web.

Combinatorics represents a better mathematical foundation for students interested in computer science and software engineering, since the majority of software concepts are taken from this field. Additionally, most systems integration concepts are more readily taught through combinatorics than other advanced mathematics, such as calculus. The concepts of graphs, sets, and formal languages all represent fundamental concepts of computer science and software.

The n-queens problem consists of placing n nonattacking queens on an $n \times n$ chessboard, and has been studied since 1869 [32]. Despite its age, the n-queens problem has served as a continued research focus in areas ranging from information theory to quantum mechanics to computer science. No current compilation exists, however, of all known n-queens related discoveries.

Top-Level Information Architecture Identification

For the n-queens.com website, the top-level information architecture consists of a conceptual block naming the website (Figure 1). This focuses the development effort on web-based materials, but is not necessarily clear enough to allow easy system elaboration. To enable system elaboration, therefore, a brief overview description is necessary.

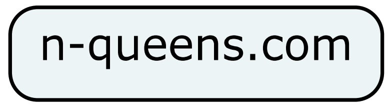

Figure 1. The top-level information architecture of n-queens.com simply names the website.

The n-queens.com website has as it's general goals combinatorics education using the N-Queens problem; teaching the concepts of process, system, design, and metric; and serving as the definitive resource for n-queens material on the web.

System Elaboration

The majority of content of an informational website can be grouped into articles. These articles may have images, videos, and interactive components, but are generally self-contained. For the n-queens.com website, the second system elaboration strategy is used, where as many sub-components of the overall system as possible are identified, as in Figure 2.

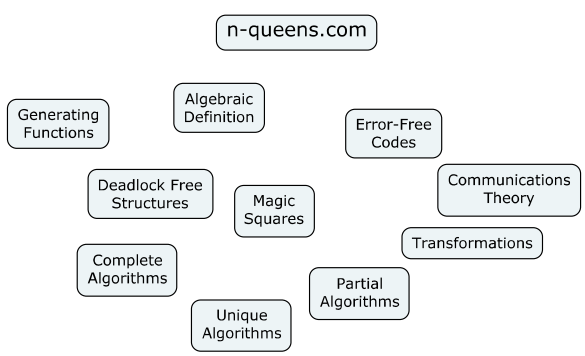

Figure 2. Initial identification of subcomponents.

These sub-components are then organized within the concept map (Figure 3). For n-queens.com, three overall areas may be readily identified: definitions, solutions, and applications of the n-queens problem.

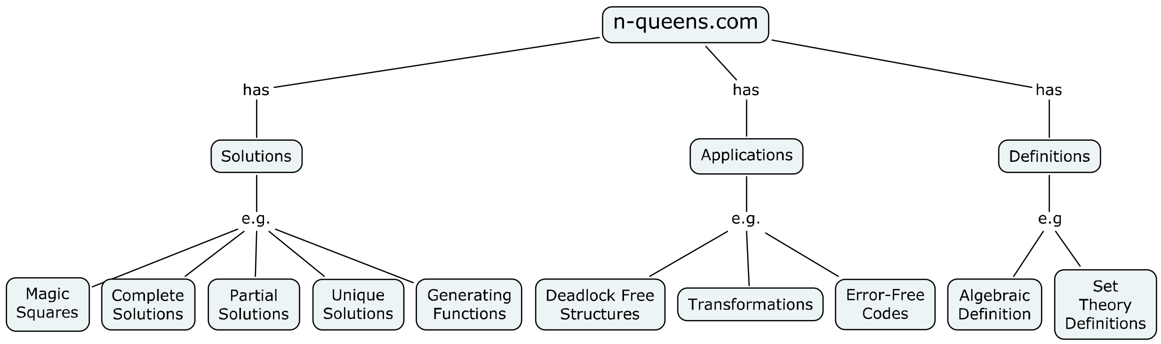

Figure 3. Initial relationships between subcomponents.

Layered Detailing

Once the initial organization is completed, the subcomponents may be further grouped under more detailed parents (Figure 4). These mid-level subcomponents further refine the information architecture by providing more detailed classifications and pointing to other potential subcomponents. As an example, classifying the error-free codes block under communication theory raises a question as to what other applications may exist in that area.

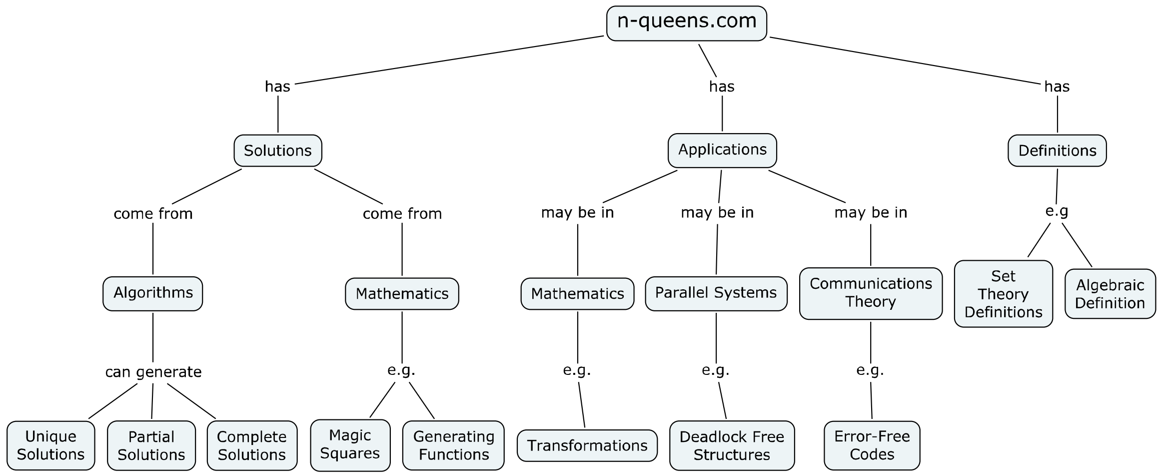

Figure 4. Detailed relationships between subcomponents.

Other areas of the n-queens.com website may also be identified by further grouping all current subcomponents under a still higher-level component, articles. Doing so allows the addition of other content, such as an n-queens workstation allowing experiments with different generating algorithms and display modalities (Figure 5).

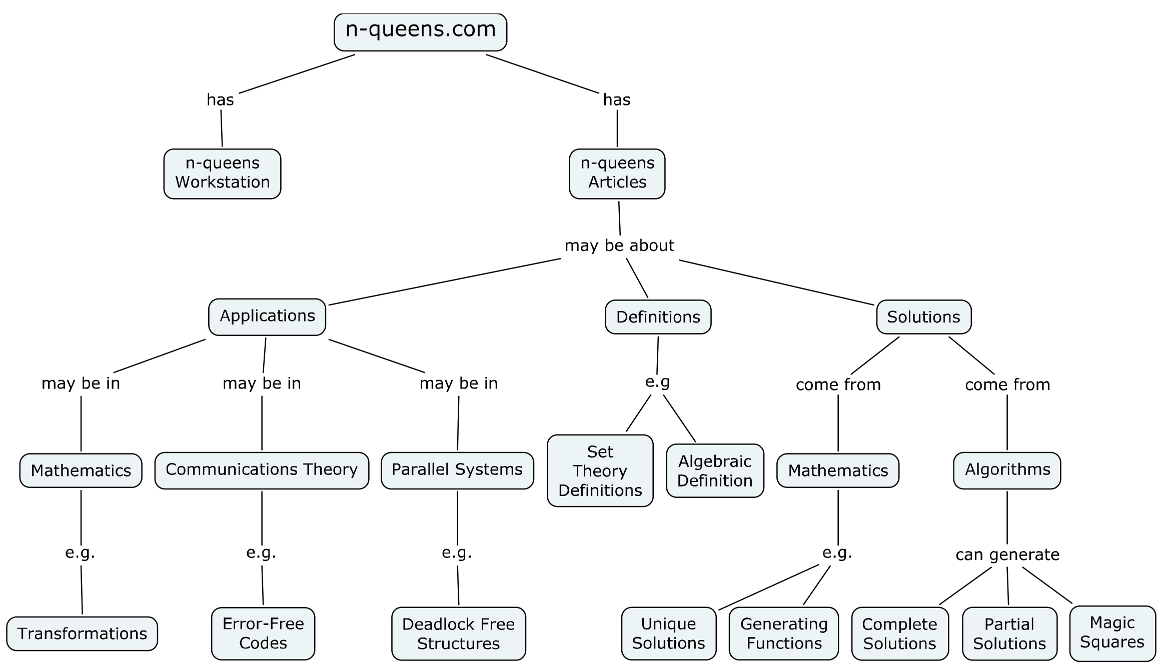

Figure 5. Further n-queens.com elaboration, including an n-queens workstation.

Clinical Decision Support System

Introduction

Recent healthcare reform laws in the United States provide strong incentives encouraging the use of Electronic Healthcare Records (EHR), as well as penalties for failing to use them [33]. Additionally, as the quantity of medical patient data available to physicians increases, EHR systems provide more reliable and better quality healthcare [34]. These factors combine to make the adoption of EHR over paper-based systems inevitable. As the use of EHR increases, healthcare providers should be expected to make the best possible use of the wealth of computable patient data EHR systems will contain.

Simultaneously, as new advances in preventive medicine and treatment of existing conditions are discovered, processes for incorporating that knowledge into the medical practice at large are required. This incorporation will ensure appropriate recommendations, prescriptions, diagnoses, and procedures are applied in a cost effective manner.

Both the optimal use of EHR data and the rapid incorporation of new medical research can be accomplished with a Clinical Decision Support System (CDSS).

Top-Level Information Architecture Identification

The top-level information architecture of a CDSS system consists simply of a concept block identifying and naming the system. Since a well-known name already exists for clinical decision support systems, this name is used (Figure 6). A description is then developed based on the literature of the field.

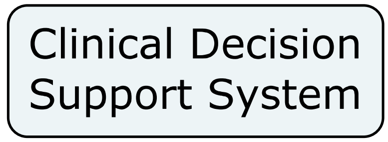

Figure 6. Top-level information architecture of a clinical decision support system.

CDSS are defined as systems providing context sensitive, computer generated, clinical and patient information for the purpose of enhancing patient care [35]. Context sensitivity requires intelligent filtering and presentation of information at appropriate times. The presentation of information by a CDSS to a physician, as described above, is called a CDSS intervention.

Initial research into the effectiveness of CDSS found them to be effective for preventive care and drug dosing, but not for diagnosis [36]. In recent years, CDSS diagnostic applications have also become effective [37]. However, the effect of CDSS on patient outcomes remains understudied and inconclusive [38]. Despite the lack of overall patient outcome conclusions, researchers have identified several factors common among successful CDSS. Table 1 summarizes these CDSS success factors.

Table 1. Success Factors for Clinical Decision Support Systems.

Success Factor	**Description**
Clinical Workflow Integration [37,39]	CDSS must be integrated into complete patient care workflow. Patient data should be entered as it is collected; interventions should be provided at the time and place of decision. For workflow integration to be successful, CDSS must also be fast. Real-time responses are required.
Simple, Minimalist Interventions [39]	Information provided to clinical decision makers should consist of brief imperative sentences. A guideline should fill, at most, a single screen. References to full guidelines and supporting information may be provided, but the initial recommendations must be short and readily understood. When additional information is

Success Factor	Description
	necessary for the CDSS to provide a recommendation, only the data absolutely needed should be requested.
Recommendation Based Interventions [37,39]	Physicians have been shown to resist stopping, even when the course of treatment they are pursuing is counterproductive. CDSS are most effective when interventions redirect rather than stop a course of treatment. Additionally, CDSS should provide recommendations for treatments rather than exclusively diagnostic assessments.
Computer-based Decision Support [37]	A CDSS is implemented as a computerized information system, rather than a collection of manual audit processes.
Continual Knowledge Base and User Interface Updates [39]	The diagnostic and treatment information in a CDSS needs frequent updating as new medical knowledge is generated by research and data-mining. Additionally, as information regarding clinician response to interventions becomes available, the presentation of those interventions should be adjusted to ensure appropriate responses. A CDSS should neither under-intervene nor over-intervene. If many needless interventions are provided, clinicians become accustomed to ignoring all interventions provided by the CDSS.

System Elaboration

At the top-most level, a CDSS takes as input patient information and provides treatment recommendations. Patient information consists of measured health parameters, patient history, and the desired outcomes. Treatment recommendations consist of single or multiple step processes for achieving the desired outcomes. Entering patient data and receiving treatment recommendations represent the fundamental use cases of a CDSS.

To accomplish this, the CDSS must contain patient information in a computable form, a treatment library, and a method for correlating treatment to desired outcomes. Ensuring that the patient information model and the treatment library remain up-to-date with current medical knowledge requires the addition of a knowledge base that collects information from publications and data mining efforts for use in the CDSS. These represent the four basic elements of a CDSS: patient model, treatment library, intelligent agents, and an authenticated knowledge base (Figure 7).

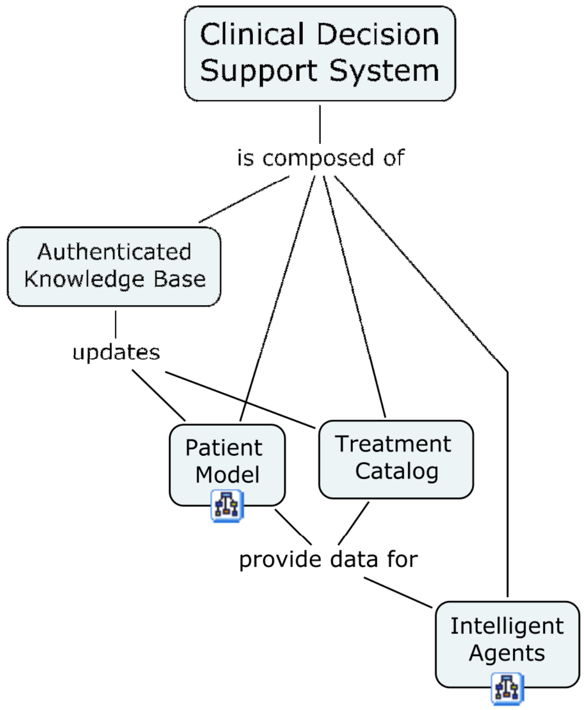

Figure 7. System elaboration of the top-level information architecture of a clinical decision support system.

Layered Detailing

Patient model. The heart of a CDSS is its patient model. Patients are the key element of any medical practice; without patients the medical profession would not exist. The view of a patient model presented here consists of a set of measured health parameters, a treatment history, and a list of desired outcomes or health goals. The measured health parameters represent the results of tests and measurements taken from the patient, including everything from blood pressure and body weight to gene sequencing. Treatment history includes all previous treatments and procedures the patient has undergone. Figure 8 presents the information architecture of a patient model.

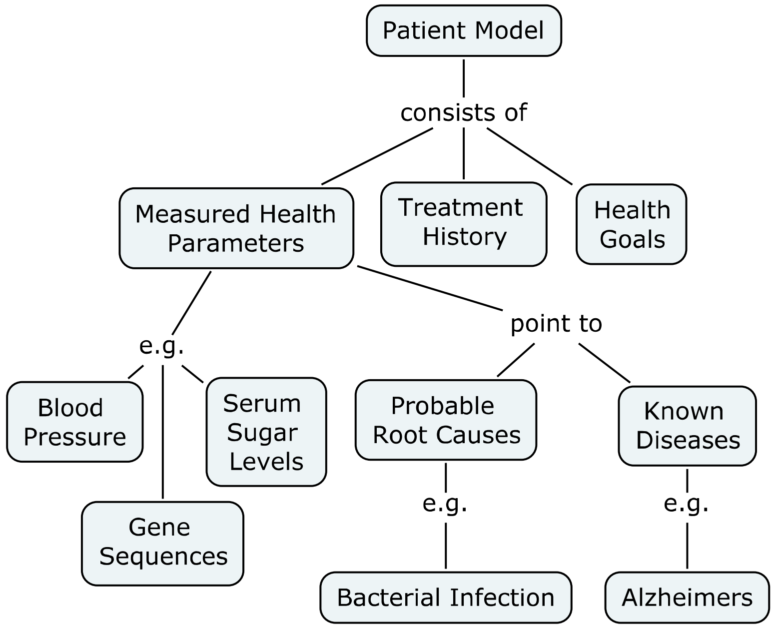

Figure 8. Information architecture of a clinical decision support system's patient model.

Measured health parameters may take a variety of forms: discrete, binary or continuous numerical values will be the most common. Ideally, a history for each parameter will also be available, allowing the system to be aware of trends and changes over time. As the CDSS amasses patient data for a large number of patients, statistical overlays may also be used to indicate whether a patient's parameters are in a normal or abnormal range.

Advances in image processing technologies will also allow image based patient data to be included, such as x-rays, MRI's, etc. Taken in summary, the measured parameters may be used to point to various disease or injury conditions. In cases where a precise disease is known to be present, a binary parameter may be used to represent that information. Where the cause of abnormal parameters is unknown, a probabilistic analysis of potential root causes may be presented. This probabilistic analysis, in turn, points to additional tests or treatments that may be necessary to further refine a diagnosis.

Treatment library. The treatment library simply consists of a list of treatments with their associated procedures, variables, and set of rules used to determine the probable effect of the treatment on a given patient's health parameters. Treatments may be flagged as safe or unsafe given certain patient conditions. In general, a treatment simply adjusts the values of measured health parameters in the patient model up or down. For instance, a treatment consisting of dieting or exercise is likely to decrease the patient's weight. Complex treatments, such as chemotherapy, affect a wide range of parameters in both positive and negative ways. The interaction between a variety of treatments, or a roadmap for a treatment schedule, requires a complex analysis of the probabilistic affects on patient parameters. Intelligent agents perform this analysis.

Intelligent agents. Intelligent agents use inference techniques to optimize a treatment or collection of treatments in order to find the minimum value of a cost function (Figure 9). The cost function, in turn, is based on health goals provided to the CDSS, the patient's information, and the treatment library.

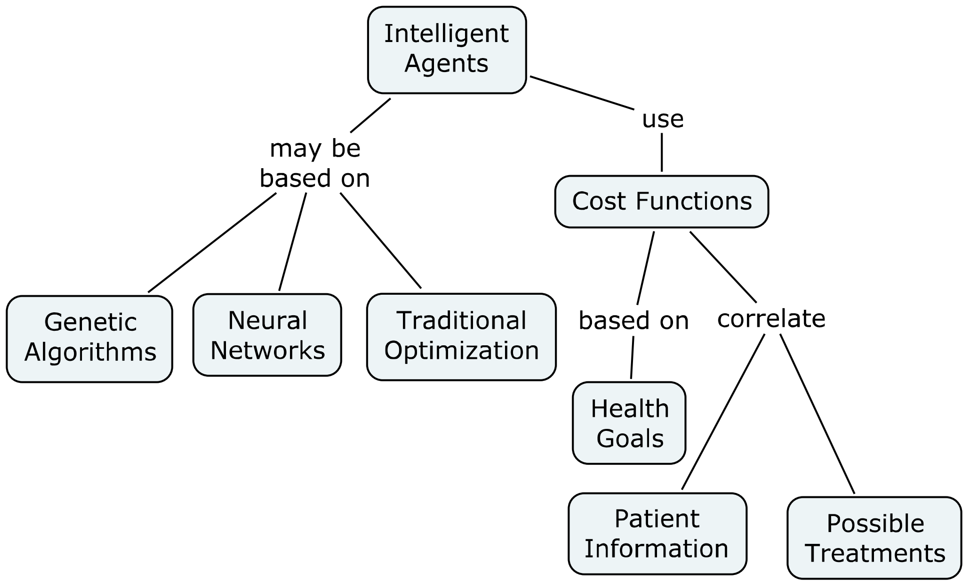

Figure 9. Information architecture of a clinical decision support system's intelligent agents.

Because the search space defined by these potentially complex cost functions is likely to be expansive, a variety of intelligent agents will generally work together. Those based on genetic or evolutionary algorithms will perform the initial partitioning of the space into valid and invalid regions, while neural network and traditional optimization based agents can further refine those zones into the optimal treatment recommendations.

Authenticated knowledge base. The information in the authenticated knowledge base is used to update both the patient models (with newly discovered parameters and relationships between parameters and disease or injury conditions) and the treatment library (with new treatments, new side effects, and new uses for old treatments). In turn, the knowledge is provided by the scientific community through journal and conference publications, as well as through data-mining of existing electronic health data.

The authenticated knowledge base represents a key advantage of a computerized CDSS over traditional health professionals. Delivery of new knowledge generated by global advances in healthcare can take years to reach the medical community in a way that is usable and reliable. Doctors in remote regions who may be critically uninformed of recent advances and discoveries may be able to improve both the efficiency and efficacy of their current practices by referring to parallel recommendations by a reliable and networked CDSS. Such an authenticated system can provide treatment recommendations for both preventative and integrative health, in addition to point of care delivery devoid of pharmaceutical bias.

Conclusion

An information architecture of a CDSS system was developed. A CDSS was found to fundamentally consist of four elements: a patient model, a treatment library, and authenticated knowledge base, and a collection of intelligent agents. In general use, patient data including measured health parameters, treatment history, and health goals is provided through the patient model to the intelligent agents. The patient's health goals form the basis for cost functions allowing the intelligent agents to optimize a course of treatments from the treatment library, and provide a recommendation back to the healthcare provider. Both the patient model and treatment library are kept up to date with respect to current medical knowledge by an authenticated knowledge base.

Note that further layered detailing may be required to enable direct parameter identification for traditional design space exploration. However, the top and mid-level information architecture identification provides a framework for the identification of those parameters and their relationships. For many unstructured problems, the emphasis in design space decomposition is not necessarily identifying parameters of interest, but identifying the overall information architecture of a solution to the problem.

DISCUSSION AND CONCLUSION

By decomposing a design space using concept maps, the proposed techniques capture the semantics of a design. This decomposition process comprises identifying the top-level information architecture, subsystem elaboration, and layered detailing until the desired level of abstraction is reached.

Furthermore, the use of software tools for the creation of concept maps enable a semantically captured design to be acted upon by other software systems. As such, it provides a method for translating designs into the semantic web.

The proposed techniques were demonstrated using two case studies: development of an n-queens based combinatorics education website, and drafting the information architecture of a clinical decision support system. These case studies showed the ability of a concept map to capture an information architecture by decomposing a design space. They also successfully demonstrate the use of computerized concept maps for generating a semantic web representation of a decomposition.

LIST OF REFERENCES

[1] H.A. Simon, *The Sciences of the Artificial*, Cambridge: The MIT Press, 1996.

[2] R.C. Conant, "Detecting subsystems of a complex system," *IEEE Trans. Syst. Man Cybern.*, vol. 2, 1972, pp. 550-553.

[3] Y. Tuncer, M.M. Tanik, and D.B. Allison, "An overview of statistical decomposition techniques applied to complex systems," *Comp. Stat. & Data Anal.*, vol. 52, Jan. 2008, pp. 2292-2310.

[4] B. Boehm and V.R. Basili, "Gaining intellectual control of software development," *Computer*, vol. 33, May. 2000, pp. 27-33.

[5] S. Colwell (10, April 1996). Intellectual Control [Online]. Available: http://stevecolwell.com/ic.html

[6] A.H. Dogru and M.M. Tanik, "A process model for component-oriented software engineering," *IEEE Softw.*, vol. 20, Mar. 2003, pp. 34-41.

[7] M. Aksit and F. Marcelloni, "Deferring elimination of design alternatives in object-oriented methods," *Concurrency and Computation: Practice and Experience*, vol. 13, Dec. 2001, pp. 1247-1279.

[8] M. M. Tanik, University of Alabama at Birmingham, Birmingham, AL, private communication, July 2010.

[9] B. Chandrasekaran, "Design problem solving: a task analysis," *AI Mag.*, vol. 11, 1990, pp. 59-71.

[10] V. Goel and P. Pirolli, "Motivating the notion of generic design within information-processing theory: the design problem space," *AI Mag.*, vol. 10, Jan. 1989, pp. 18-36.

[11] J. Juang and R. Pappa, "Eigensystem realization algorithm for modal parameter identification and model reduction," *J. of Guidance*, vol. 8, Jan. 1985, p. 620.

[12] S. Padula, C. Sandridge, and R. Haftka, "Demonstration of decomposition and optimization in the design of experimental space systems," *Recent Advances in*

Multidisciplinary Analysis and Optimization, S. Padula, C.A. Sandridge, R.T. Haftka, and J.L. Walsh, Ed., 1989, pp. 297-316.

[13] I.C. Parmee and M.A. Beck, “An evolutionary, agent-assisted strategy for conceptual design space decomposition,” *Lecture Notes in Computer Science*, D. Corne and J.L. Shapiro, Ed., Manchester, UK: Springer, 1997, pp. 275-286.

[14] A. Thompson, P. Layzell, and R.S. Zebulum, “Explorations in design space: unconventional electronics design through artificial evolution,” *IEEE Trans. Evol. Comput.*, vol. 3, Sep. 1999, pp. 167-196.

[15] I.C. Parmee, “Evolutionary design search, exploration and optimisation,” *Proc. of GECCO ’08*, New York, New York, USA: ACM Press, 2008, p. 2777.

[16] S.G. Abraham, B.R. Rau, and R. Schreiber, (28, July 2000) Fast design space exploration through validity and quality filtering of subsystem designs [Online]. Available: http://www.hpl.hp.com/techreports/2000/HPL-2000-98.html

[17] L. Nigay and J. Coutaz, “A design space for multimodal systems,” *Proc. Conf. on Human Factors in Comput. Sys.*, New York, New York, USA: ACM Press, 1993, pp. 172-178.

[18] J. Moses, “The anatomy of large scale systems,” *MIT Engineering Systems Division Working Paper Series*, MIT, 2003.

[19] J. Moses, “Three design methodologies, their associated organizational structures and their relationship to various fields,” *MIT Engineering Systems Division Working Paper Series*, MIT, 2004.

[20] J.M. Sussman, “Ideas on complexity in systems -- twenty views,” *MIT Engineering Systems Division Working Paper Series*, MIT, 2000.

[21] J.M. Sussman, “Collected views on complexity in systems,” *MIT Engineering Systems Division Working Paper Series*, MIT, 2003.

[22] I. Parmee and J. Abraham, “Interactive evolutionary design,” *Knowledge Incorporation in Evolutionary Computation*, Y. Jin, Ed., Springer, 2005, pp. 435-458.

[23] J. Novak and A. Cañas, *The theory underlying concept maps and how to construct and use them*, Pensacola, FL: Florida Institute for Human and Machine Cognition, 2008.

[24] M.A. Ruiz-Primo and R.J. Shavelson, “Problems and issues in the use of concept maps in science assessment,” *Journal of Research in Science Teaching*, vol. 33, Oct. 1996, pp. 569-600.

[25] T. Sumner, F. Ahmad, S. Bhushan, Q. Gu, F. Molina, S. Willard, M. Wright, L. Davis, and G. Janee, “Linking learning goals and educational resources through interactive concept map visualizations,” *International Journal on Digital Libraries*, vol. 5, Aug. 2004, pp. 18-24.

[26] Y. Yin, J. Vanides, M.A. Ruiz-Primo, C.C. Ayala, and R.J. Shavelson, “Comparison of two concept-mapping techniques: Implications for scoring, interpretation, and use,” *Journal of Research in Science Teaching*, vol. 42, Feb. 2005, pp. 166-184.

[27] V.P. Gurupur, “Building applications representing processes as concepts,” *SDPS 2010*, Society for Design and Process Science, Dallas, TX, June 2010.

[28] D.E. Robbins and J. Tanik, “Exploring frontiers in e-learning technologies: a rapid prototyping portal approach for design and process methodologies,” *Twelfth Transdisciplinary Conference-Workshop on Integrated Design and Process Science*, Montgomery, AL, November 2009.

[29] D.E. Robbins, *N-queens.com framework*, unpublished.

[30] A. Canas, G. Hill, R. Carff, N. Suri, and J. Lott, “CmapTools: A knowledge modeling and sharing environment,” *Proc. of the First Int. Conference on Concept Mapping*, Jan. 2004.

[31] V.P. Gurupur and M.M. Tanik, “A system for building clinical research applications using semantic web-based approach,” *J. of Medical Syst.,* Feb. 2010.

[32] J. Bell and B. Stevens, “A survey of known results and research areas for n-queens,” *Discrete Mathematics*, vol. 309, Jan. 2009, pp. 1-31.

[33] Health Information Technology for Economic and Clinical Health (HITECH) Act, Public Law No. 111-5, Title XIII, 2009, p. 226-279. Available: http://www.gpo.gov/fdsys/pkg/PLAW-111publ5/pdf/PLAW-111publ5.pdf

[34] L.C. Burton, G.F. Anderson, and I.W. Kues, “Using electronic health records to help coordinate care,” *The Milbank Quarterly*, vol. 82, Jan. 2004, pp. 457-81.

[35] J.A. Osheroff, E.A. Pifer, J.M. Teich, D.F. Sittig, and R.A. Jenders, *Improving Outcomes with Clincal Decision Support: An Implementer’s Guide*, Chicago: HIMSS, 2005.

[36] D.L. Hunt, R.B. Haynes, S.E. Hanna, and K. Smith, “Effects of computer-based clinical decision support systems on physician performance and patient outcomes: a systematic review,” *JAMA*, vol. 280, Oct. 1998, pp. 1339-46.

[37] K. Kawamoto, C.A. Houlihan, E.A. Balas, and D.F. Lobach, “Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success,” *BMJ,* vol. 330, Apr. 2005, p. 765.

[38] A.X. Garg, N.K.J. Adhikari, H. McDonald, M.P. Rosas-Arellano, P.J. Devereaux, J. Beyene, J. Sam, and R.B. Haynes, “Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: a systematic review,” *JAMA*, vol. 293, Mar. 2005, pp. 1223-38.

[39] R.A.K. Horasani, M.I.T. Anasijevic, B.L.M. Iddleton, and M.S. C, “Ten commandments for effective clinical decision support : making the practice of evidence-based medicine a reality,” *J. of the American Medical Informatics Association*, vol. 10, 2003, pp. 523-530.

APPENDIX: CASE STUDIES AS SEMANTIC TRIPLES

The final, elaborated concept map for each case study was exported as a semantic web file in the N3-Tripple format. The resulting semantic data is presented here.

N-queens.com Semantic Content

```
@prefix xsd:     <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs:    <http://www.w3.org/2000/01/rdf-schema#> .
@prefix daml:    <http://www.daml.org/2001/03/daml+oil#> .
@prefix rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl:     <http://www.w3.org/2002/07/owl#> .
@prefix :        <http://localhost/default#> .

:Definitions <http://localhost/default#e.g>
<http://localhost/default#Set%20Theory%20Definitions> .
:Definitions <http://localhost/default#e.g>
<http://localhost/default#Algebraic%20Definition> .
:Mathematics <http://localhost/default#e.g.> :Transformations .
:Mathematics <http://localhost/default#e.g.>
<http://localhost/default#Generating%20Functions> .
:Mathematics <http://localhost/default#e.g.>
<http://localhost/default#Unique%20Solutions> .
<http://localhost/default#n-queens%20Articles>
<http://localhost/default#may%20be%20about> :Applications .
<http://localhost/default#n-queens%20Articles>
<http://localhost/default#may%20be%20about> :Solutions .
<http://localhost/default#n-queens%20Articles>
<http://localhost/default#may%20be%20about> :Definitions .
:Algorithms <http://localhost/default#can%20generate>
<http://localhost/default#Complete%20Solutions> .
:Algorithms <http://localhost/default#can%20generate>
<http://localhost/default#Partial%20Solutions> .
:Algorithms <http://localhost/default#can%20generate>
<http://localhost/default#Magic%20Squares> .
:Solutions <http://localhost/default#come%20from> :Algorithms .
:Solutions <http://localhost/default#come%20from> :Mathematics .
<http://localhost/default#n-queens.com> :has
<http://localhost/default#n-queens%20Articles> .
<http://localhost/default#n-queens.com> :has
<http://localhost/default#n-queens%20Workstation> .
<http://localhost/default#Communications%20Theory>
<http://localhost/default#e.g.> <http://localhost/default#Error-
Free%20Codes> .
:Applications <http://localhost/default#may%20be%20in> :Mathematics .
```

```
:Applications <http://localhost/default#may%20be%20in>
<http://localhost/default#Parallel%20Systems> .
:Applications <http://localhost/default#may%20be%20in>
<http://localhost/default#Communications%20Theory> .
<http://localhost/default#Parallel%20Systems>
<http://localhost/default#e.g.>
<http://localhost/default#Deadlock%20Free%20Structures> .
```

Clinical Decision Support System Semantic Content

Top Level

```
@prefix xsd:     <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs:    <http://www.w3.org/2000/01/rdf-schema#> .
@prefix daml:    <http://www.daml.org/2001/03/daml+oil#> .
@prefix rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl:     <http://www.w3.org/2002/07/owl#> .
@prefix :        <http://localhost/default#> .

<http://localhost/default#Probable%20Root%20Causes>
<http://localhost/default#e.g.>
<http://localhost/default#Bacterial%20Infection> .
:Conferences <http://localhost/default#provide%20new%20knowledge%20for>
<http://localhost/default#Authenticated%20Knowledge%20Base> .
<http://localhost/default#Intelligent%20Agents> :provides
<http://localhost/default#Real-time%20Decision%20Support> .
<http://localhost/default#Intelligent%20Agents> :provides
<http://localhost/default#Health%20Goal%20Roadmaps> .
<http://localhost/default#Intelligent%20Agents>
<http://localhost/default#may%20be%20based%20on>
<http://localhost/default#Traditional%20Optimization> .
<http://localhost/default#Intelligent%20Agents>
<http://localhost/default#may%20be%20based%20on>
<http://localhost/default#Neural%20Networks> .
<http://localhost/default#Intelligent%20Agents>
<http://localhost/default#may%20be%20based%20on>
<http://localhost/default#Genetic%20Algorithms> .
<http://localhost/default#Intelligent%20Agents> :use
<http://localhost/default#Cost%20Functions> .
<http://localhost/default#David%20Robbins>
<http://localhost/default#has%20studied>
<http://localhost/default#Design%20Space%20Decomposition%20using%20Conc
ept%20Maps> .
<http://localhost/default#David%20Robbins>
<http://localhost/default#has%20studied>
<http://localhost/default#Concept%20Maps> .
<http://localhost/default#David%20Robbins> a
<http://localhost/default#Computer%20Engineering%20Graduate%20Student>  .
:Datamining <http://localhost/default#provide%20new%20knowledge%20for>
<http://localhost/default#Authenticated%20Knowledge%20Base> .
<http://localhost/default#Cost%20Functions>
<http://localhost/default#based%20on>
<http://localhost/default#Health%20Goals> .
<http://localhost/default#Cost%20Functions> :correlate
<http://localhost/default#Possible%20Treatments> .
```

```
<http://localhost/default#Cost%20Functions> :correlate
<http://localhost/default#Patient%20Information> .
<http://localhost/default#Authenticated%20Knowledge%20Base> :updates
<http://localhost/default#Patient%20Model> .
<http://localhost/default#Authenticated%20Knowledge%20Base> :updates
<http://localhost/default#Treatment%20Catalog> .
<http://localhost/default#Measured%20Health%20Parameters>
<http://localhost/default#e.g.>
<http://localhost/default#Blood%20Pressure> .
<http://localhost/default#Measured%20Health%20Parameters>
<http://localhost/default#e.g.>
<http://localhost/default#Gene%20Sequences> .
<http://localhost/default#Measured%20Health%20Parameters>
<http://localhost/default#e.g.>
<http://localhost/default#Serum%20Sugar%20Levels> .
<http://localhost/default#Measured%20Health%20Parameters>
<http://localhost/default#can%20be> :Continuous .
<http://localhost/default#Measured%20Health%20Parameters>
<http://localhost/default#can%20be> :Discrete .
<http://localhost/default#Measured%20Health%20Parameters>
<http://localhost/default#can%20be> :Binary .
<http://localhost/default#Measured%20Health%20Parameters>
<http://localhost/default#point%20to>
<http://localhost/default#Probable%20Root%20Causes> .
<http://localhost/default#Measured%20Health%20Parameters>
<http://localhost/default#point%20to>
<http://localhost/default#Known%20Diseases> .
:Journals <http://localhost/default#provide%20new%20knowledge%20for>
<http://localhost/default#Authenticated%20Knowledge%20Base> .
<http://localhost/default#Clinical%20Decision%20Support%20System>
<http://localhost/default#is%20composed%20of>
<http://localhost/default#Treatment%20Catalog> .
<http://localhost/default#Clinical%20Decision%20Support%20System> :prov
ides <http://localhost/default#Health%20Goal%20Roadmaps> .
<http://localhost/default#Clinical%20Decision%20Support%20System>
<http://localhost/default#can%20be%20applied%20to> :Veterinary .
<http://localhost/default#Clinical%20Decision%20Support%20System>
<http://localhost/default#is%20composed%20of>
<http://localhost/default#Authenticated%20Knowledge%20Base> .
<http://localhost/default#Clinical%20Decision%20Support%20System>
<http://localhost/default#can%20be%20applied%20to>
<http://localhost/default#International%20Space%20%20Station> .
<http://localhost/default#Clinical%20Decision%20Support%20System>
<http://localhost/default#is%20composed%20of>
<http://localhost/default#Patient%20Model> .
<http://localhost/default#Clinical%20Decision%20Support%20System>
<http://localhost/default#is%20being%20information%20architected>
<http://localhost/default#David%20Robbins> .
<http://localhost/default#Clinical%20Decision%20Support%20System>
<http://localhost/default#can%20be%20applied%20to>
<http://localhost/default#Sport%20Medicine> .
<http://localhost/default#Clinical%20Decision%20Support%20System>
<http://localhost/default#is%20composed%20of>
<http://localhost/default#Intelligent%20Agents> .
<http://localhost/default#Clinical%20Decision%20Support%20System>
<http://localhost/default#will%20be%20developed%20according%20to>
<http://localhost/default#CDSS%20Development%20Roadmap> .
```

```
<http://localhost/default#Clinical%20Decision%20Support%20System> :prov
ides <http://localhost/default#Real-time%20Decision%20Support> .
<http://localhost/default#Known%20Diseases>
<http://localhost/default#e.g.> :Alzheimers .
<http://localhost/default#Treatment%20Catalog>
<http://localhost/default#provide%20data%20for>
<http://localhost/default#Intelligent%20Agents> .
<http://localhost/default#Computer%20Engineering%20Graduate%20Student>
:at      :UAB .
<http://localhost/default#Patient%20Model>
<http://localhost/default#consists%20of>
<http://localhost/default#Measured%20Health%20Parameters> .
<http://localhost/default#Patient%20Model>
<http://localhost/default#consists%20of>
<http://localhost/default#Treatment%20History> .
<http://localhost/default#Patient%20Model>
<http://localhost/default#provide%20data%20for>
<http://localhost/default#Intelligent%20Agents> .
```

Patient Model Detail

```
@prefix xsd:     <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs:    <http://www.w3.org/2000/01/rdf-schema#> .
@prefix daml:    <http://www.daml.org/2001/03/daml+oil#> .
@prefix rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl:     <http://www.w3.org/2002/07/owl#> .
@prefix :        <http://localhost/default#> .

<http://localhost/default#Known%20Diseases>
<http://localhost/default#e.g.> :Alzheimers .
<http://localhost/default#Measured%20Health%20Parameters>
<http://localhost/default#can%20be> :Discrete .
<http://localhost/default#Measured%20Health%20Parameters>
<http://localhost/default#can%20be> :Continuous .
<http://localhost/default#Measured%20Health%20Parameters>
<http://localhost/default#can%20be> :Binary .
<http://localhost/default#Measured%20Health%20Parameters>
<http://localhost/default#e.g.>
<http://localhost/default#Serum%20Sugar%20%20Levels> .
<http://localhost/default#Measured%20Health%20Parameters>
<http://localhost/default#e.g.>
<http://localhost/default#Gene%20Sequences> .
<http://localhost/default#Measured%20Health%20Parameters>
<http://localhost/default#e.g.>
<http://localhost/default#Blood%20Pressure> .
<http://localhost/default#Measured%20Health%20Parameters>
<http://localhost/default#point%20to>
<http://localhost/default#Known%20Diseases> .
<http://localhost/default#Measured%20Health%20Parameters>
<http://localhost/default#point%20to>
<http://localhost/default#Probable%20Root%20Causes> .
<http://localhost/default#Probable%20Root%20Causes>
<http://localhost/default#e.g.>
<http://localhost/default#Bacterial%20Infection> .
```

```
<http://localhost/default#Patient%20Model>
<http://localhost/default#consists%20of>
<http://localhost/default#Health%20Goals> .
<http://localhost/default#Patient%20Model>
<http://localhost/default#consists%20of>
<http://localhost/default#Treatment%20History> .
<http://localhost/default#Patient%20Model>
<http://localhost/default#consists%20of>
<http://localhost/default#Measured%20Health%20Parameters> .
```

Intelligent Agents Detail

```
@prefix xsd:      <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs:     <http://www.w3.org/2000/01/rdf-schema#> .
@prefix daml:     <http://www.daml.org/2001/03/daml+oil#> .
@prefix rdf:      <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl:      <http://www.w3.org/2002/07/owl#> .
@prefix :         <http://localhost/default#> .

<http://localhost/default#Cost%20Functions>
<http://localhost/default#based%20on>
<http://localhost/default#Health%20Goals> .
<http://localhost/default#Cost%20Functions> :correlate
<http://localhost/default#Patient%20Information> .
<http://localhost/default#Cost%20Functions> :correlate
<http://localhost/default#Possible%20Treatments> .
<http://localhost/default#Intelligent%20Agents>
<http://localhost/default#may%20be%20based%20on>
<http://localhost/default#Genetic%20Algorithms> .
<http://localhost/default#Intelligent%20Agents>
<http://localhost/default#may%20be%20based%20on>
<http://localhost/default#Neural%20Networks> .
<http://localhost/default#Intelligent%20Agents>
<http://localhost/default#may%20be%20based%20on>
<http://localhost/default#Traditional%20Optimization> .
<http://localhost/default#Intelligent%20Agents> :use
<http://localhost/default#Cost%20Functions> .
```