2010

A Framework for Composite Service Development: Process-as-a-Concept

Varadraj Prabhu Gurupur
University of Alabama at Birmingham

A FRAMEWORK FOR COMPOSITE SERVICE DEVELOPMENT: PROCESS-AS-A-CONCEPT

by

VARADRAJ PRABHU GURUPUR

MURAT M. TANIK, CHAIR
B. EARL WELLS
GREGG L. VAUGHN
MURAT N. TANJU
SANJAY K. SINGH

A DISSERTATION

Submitted to the graduate faculty of The University of Alabama at Birmingham,
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

BIRMINGHAM, ALABAMA

2010

A FRAMEWORK FOR COMPOSITE SERVICE DEVELOPMENT:
PROCESS-AS-A-CONCEPT

VARADRAJ PRABHU GURUPUR

COMPUTER ENGINEERING

ABSTRACT

This dissertation addresses the problem of bridging the gap that exists between user requirements and the implemented software solution. This gap between the user needs and the actual implementation is usually termed the semantic gap problem. In this dissertation, the software solution is viewed as a composite of available software services and seeking semantically motivated approaches for the composition of a final solution. Representing user needs by capturing the semantic components of these needs and using this representation to build a composite service solution are therefore critical in reducing the aforementioned semantic gap.

Although scientists have presented solutions that contribute to solving the problem of semantic gap, these solutions individually have not adequately resolved this problem. This dissertation presents a combinatorial approach in the form of a framework for reducing this semantic gap in building composite services. We term this framework the Process-as-a-Concept (PAAC) Framework because it is based on the philosophy of converting a process representation into a concept.

The proposed PAAC Framework, designed to facilitate the reduction of this semantic gap, provides a structured guiding path for the semantic service composition process by using the input attributes provided by the user. The composition process accepts the user needs, employed to identify the user processes. The user processes are then represented in the form of a task system model. The task system model is then

converted into a concept map, which is again converted into an XML-based document. The document thus obtained is in a machine-actable format that can be easily used to select appropriate services with which to build the composite service.

The dissertation compares the PAAC Framework with other existing solutions that can be useful in reducing the problem of semantic gap. This framework can be further extended to include the conversion of the representation of a system into a concept map. Overall, we have introduced the idea of representing a process in the form of a concept and thereby reducing the semantic gap involved in building composite services.

DEDICATION

I dedicate this dissertation to my wife, Sarayu, and to my brother, Damodar.

ACKNOWLEDGMENT

First and foremost, I thank my advisor, Dr. Murat M. Tanik of the Department of Electrical and Computer Engineering (ECE) at the University of Alabama at Birmingham (UAB), for his invaluable advice, kind assistance, and contributions toward my dissertation. He encouraged me to be a good researcher and to develop critical thinking. Without his support, I would not have been able to develop the skills required to obtain a doctoral degree.

I must express my gratitude to Dr. Murat N. Tanju of the Department of Accounting and Information Sciences at UAB for his critical evaluation of my work. I also thank my dear friend, Dr. Urcun J. Tanik for mentoring me in research activities. He trained me in developing research ideas during my early days as a student at UAB.

I take this opportunity to thank Dr. Rajani S. Sadasivam for helping me develop the idea of converting a task system model into a concept map. Also, I thank my friends and colleagues Dr. Tolga Tuncer, Dr. Ozgur Aktunc, and Bunyamin Ozaydin, for their wonderful support.

I thank Dr. B. Earl Wells and Dr. Sanjay K. Singh for helping me improve my presentation of my dissertation.

Many thanks to Dr. Gregg L. Vaughn, and Dr. Gary J. Grimes for being a part of my doctoral committee and for contributing valuable inputs.

I am also indebted to Dr. Mark Todd, Assistant Vice President of UAB hospital, for providing me with the opportunity to work in the Department of Pharmacy as a Data Processing Specailist and for encouraging me to pursue my research goals. I later became a member of the UAB Department of Neurology by Dr. H. Randall Griffith and Dr. Daniel C. Marson, who gave me the opportunity to work with the Alzheimer's Disease Research Center at UAB.

At UAB, I have had the privilege of working with a team of wonderful co-workers and supervisors. I thank my co-workers, Penny L. Forsyth, Paula S. Lowry, Paula D. Ledlow, Linda Woodard, and Sharonda Hardy. I also convey my regards to my supervisors, Katherine L. Belue, Alice Ookeditse, Sara Kryswanski, and Karen Fields, for their excellent support.

Last, I thank Dr. Alfred Bartolucci of the UAB Department of Biostatistics and Dr. David G. Clark of the UAB Department of Neurology for their valuable inputs, which has improved the quality of my publications.

TABLE OF CONTENTS

LIST OF TABLES

Table *Page*

LIST OF FIGURES

Figure *Page*

LIST OF ABBREVIATIONS

BPEL	Business Process Execution Language
BPEL4WS	Business Process Execution Language for Web Services
CommonKADS	Common Knowledge Acquisition and Design Synthesis
CP^2	Composite Process Personalization
DoDAF	Department of Defense Architecture Framework
DOGMA	Developing Ontology Grounded Methods and Applications
IHMC	Institute for Human and Machine Cognition
INCOSE	International Council on Systems Engineering
KR	Knowledge Representation
NIH	National Institutes of Health
OMG	Object Management Group
OMIM	Online Mendelian Inheritance in Man
OWL	Web Ontology Language
PAAC	Process-as-a-Concept
RDF	Resource Description Framework
SPARQL	SPARQL Protocol and RDF Query Language
SysML	Systems Modeling Language
UAB	The University of Alabama at Birmingham
UML	Unified Modeling Language

UMLS	Unified Medical Language System
USPSTF	United States Preventive Services Task Force
W3C	World Wide Web Consortium
XML	Extensible Markup Language

CHAPTER 1

INTRODUCTION

A. Problem of Semantic Gap in Software Development

A gap exists between user needs and software implementation efforts in the area of software development [1]. This problem persists mainly because software systems often do not fully support user needs [2]. This gap, termed a semantic gap (Figure 1), exists mainly for two reasons: a) lack of communication between the users and developers of the system and b) lack of integrated tool support across the different software development phases, which further leads to a breakdown in communication.

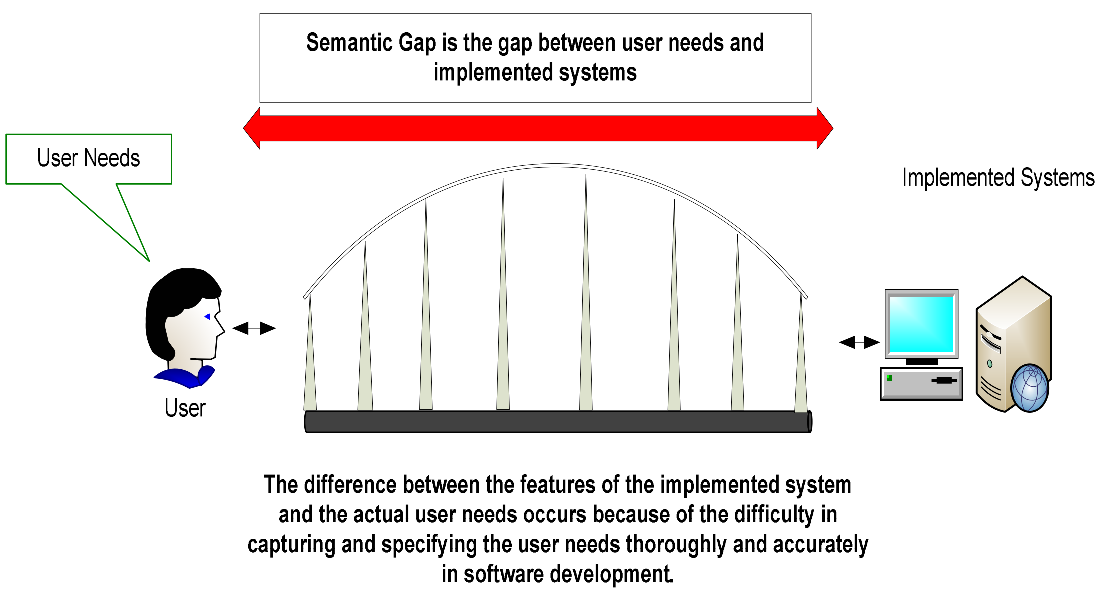


Figure 1. Overview of the semantic gap problem

Chomsky [3] states that "grammar of a language can be viewed as a theory of the structure of this language." Since it is difficult to limit the grammar of a language to a particular group of sentences the ambiguity involved in natural languages sometimes intensifies the breakdown in communication between user needs and implementation. This problem is critical because it leads to major inefficiencies in the operation of enterprises and therefore results in increased costs and low productivity. Many enterprises have failed to succeed due to software failures caused by implementations not meeting the user needs [4]. The existence of a semantic gap may have the following effects:

i) Difficulty in translating abstract information into an executable entity.

ii) Difficulty in communication between the user and developer as a result of lack of domain expertise.

iii) Increase in time required to build the implementation as per user needs.

Historically, this semantic gap problem was addressed with tools at different levels of abstraction, as depicted in Figure 2. The approach taken in this dissertation is to view the software solution as a composite of available software services and to seek semantically motivated methods for the composition of a final solution. It is our understanding that various emerging methods and technologies such as the semantic service technology, concept map tools, and information entropy can be used effectively to bridge the semantic gap. This dissertation provides a solution to the problem of the semantic gap by representing processes as concepts.

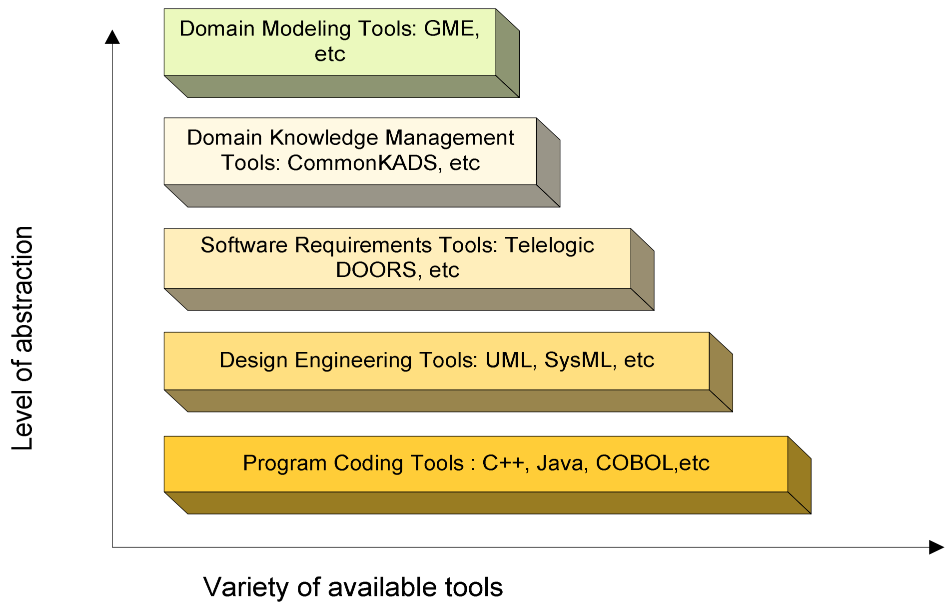


Figure 2. Variety of available tools versus level of abstraction

1) Existence of Semantic Gap in Building Medical Research Software

The problem delineated in Section A can be better understood by considering the example of the difficulty in building software for medical research [5]. This difficulty exists at least in part for the following reasons:

i) Lack of domain expertise: The clinicians are usually not software developers, and the presence of multiple terminologies makes the process of building medical research applications difficult.

ii) High cost of software development: A sizeable share of the research grant is used in building the required software application.

iii) Time involved in building the software application: A sizeable part of the grant period is used in building the software application.

iv) Communication barriers: It is difficult for the clinician to explain the terminologies to the application developer.

Figure 3 depicts a typical medical research system. The five steps shown in the figure are commonly used in a typical medical research study. The University of Alabama at Birmingham (UAB) has several medical research centers where investigators conduct research activities on various medical and biomedical processes.

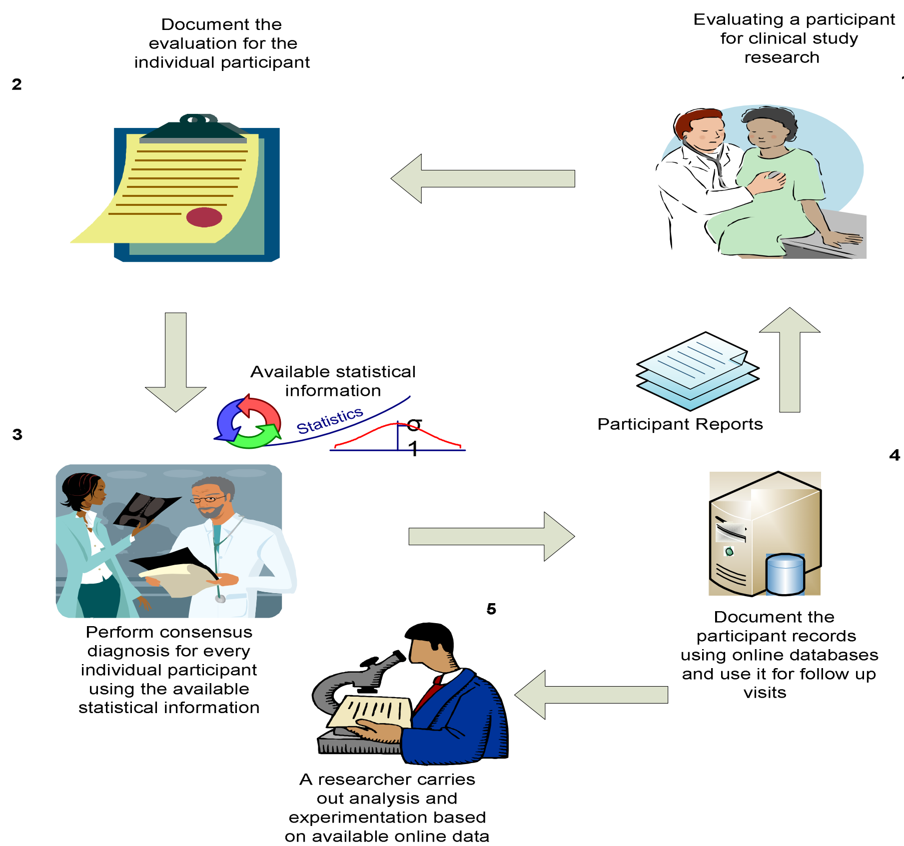


Figure 3. A typical medical research system

Note: From “A System for Building Clinical Research Applications using Semantic Web-Based Approach,” by V. Gurupur and M. M. Tanik, *Journal of Medical Systems,* DOI: 10.1007/s10916-010-9445-8. Copyright 2010 by Springer. Adapted with permission.

Over the years, students and faculty members from the UAB School of Engineering have worked with clinicians and biostatisticians to develop various medical research applications. This process of formulating various medical research applications has helped the developers from UAB School of Engineering understand the intricacies and obstacles involved in building applications for medical research. A critical hurdle encountered in building a medical research application consists of a semantic gap, which is manifested in the form of lack of communication between the researcher and the software developer. The primary reason for the existence of communication gap is the presence of medical terminologies that may not be easily understood by software developers who are not adequately exposed to the parlance of medical science.

B. Need for Integrating the Software Requirements by Bridging the Semantic Gap within the Requirements

Software requirements need to be developed in a manner that allows them to be converted appropriately into software design. Usually, the problem domain is well understood by the domain expert, who educates the Requirements Engineer about the problem domain. The Requirements Engineer uses his/her understanding of the domain to build the requirements document. It must be noted that the Requirements Engineer may not fully comprehend the problem domain, especially if he/she must deal with unfamiliar terminologies. For example, if the Requirements Engineer needs to build software for medical research, then it is imperative that he/she be well versed in some of the complex terminologies and concepts used in medical science.

The need for better integration in software design and implementation by filling the semantic gap constitutes a critical motivation for building a framework that engenders a higher level of maturity in the process of software development. Such a framework will mitigate the nature of labor intensiveness involved in the lifecycle of software development and thus provide a better integrated approach in building a software system. Recently, several approaches such as Rapid Application Development (RAD) [6] and other methods [7] have been employed in the process of developing a software system. However, these methods have mainly involved coding and do not include the automation of the process of software design. The process of developing a software system can be greatly elevated by applying to this process the principles used in the development of an enterprise system [8].

Although several approaches to integrate the process of software development were proposed, the absence of required technologies and methodologies such as Semantic Web Technology [9], SysML [10], and other useful methods and techniques [11] required to achieve this goal was mainly responsible for the difficulty in bringing this idea to fruition. The rise of component-based software development has engendered the usage of several mathematical theories [11]; axioms [12]; and methods such as Information Entropy [13], Latent Semantic Analysis [14], and other useful methods [8]. Improving the process of software development requires a framework that provides an opportunity to harness the versatility of mathematics applied to Semantic Service Composition [15].

There is a need for coalescing these methods and technologies to bring about an integrated solution to the problem of converting software requirements into design information. This solution requires that the methods and technologies be integrated into a framework for software design and development. While the framework provides the required abstract information on the tasks to be carried out at various levels of building an enterprise software system, the methods and technologies provide the implementation techniques with which a proof of concept can be developed to validate the framework.

C. Semantic Gap in Composite Service Development

The World Wide Web Consortium (W3C) [16] defined a web service as a "software system designed to support interoperable machine-to-machine interaction over a network." These individual web services can be developed to form a composite service. The process of building a composite service is depicted in Figure 4.

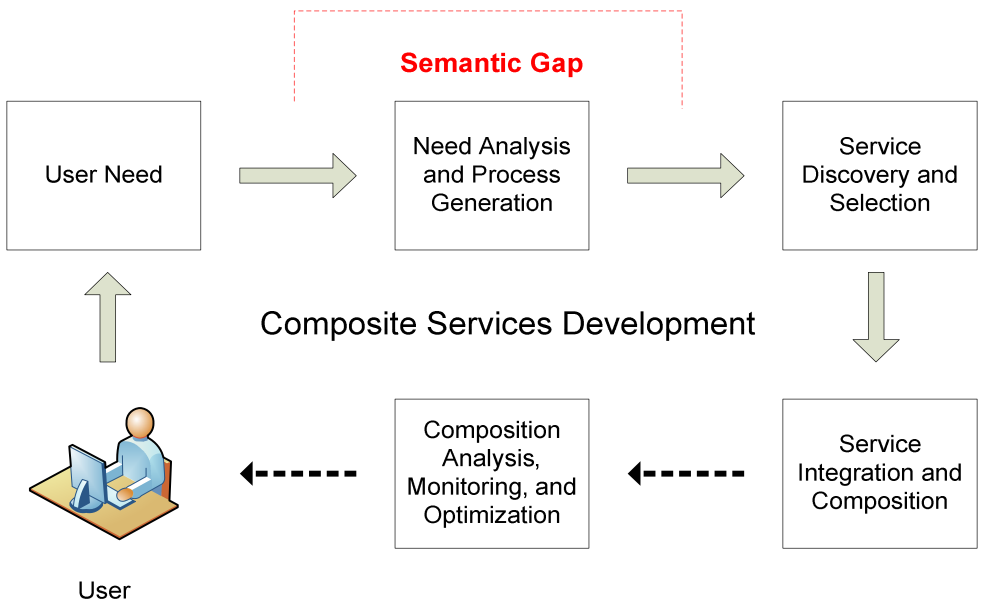


Figure 4. Semantic gap in composite service development

Dustar [17] defined the process of service composition as "those activities required to combine and link existing web services (atomic and composite services) and other components to create new processes." The development of composite services as delineated by Sadasivam [18] in his dissertation involves identifying user needs, performing need analysis, service discovery and selection, and service integration and composition, while analyzing the composite service thus obtained.

C.1 Identifying User Needs

In the initial step, the overall needs of the user must be identified in terms of the final results expected by the user.

C.2 Need Analysis and Process Generation

The needs identified by the user are decomposed into individual processes required to be executed to fulfill the user needs.

C.3 Service Discovery and Selection

Appropriate web services are identified and selected for the purpose of service composition.

C.4 Service Integration and Composition

The selected web services are integrated and composed to form composite services.

C.5 Composition Analysis, Monitoring, and Optimization

This step is used to verify and validate the composition of services. Although the process of service composition works well from the point of service selection and discovery, there exists a critical semantic gap between determining user needs and selecting the appropriate services with which to satisfy these needs.

D. Semantic Gap in Process Modeling

In his dissertation, Sadasivam [18] delineates two levels of loss of semantics in the process of building composite services (Figure 5).

i) A clear loss of semantics in converting user needs to abstract process models that occur because the process models are not sufficiently comprehensive in capturing the semantics of the user needs.

ii) A loss of semantics in converting abstract process models into executable process models that occur because the modeling technologies are inadequate in their support to configure and manage the semantics determined through the use of the abstract process models.

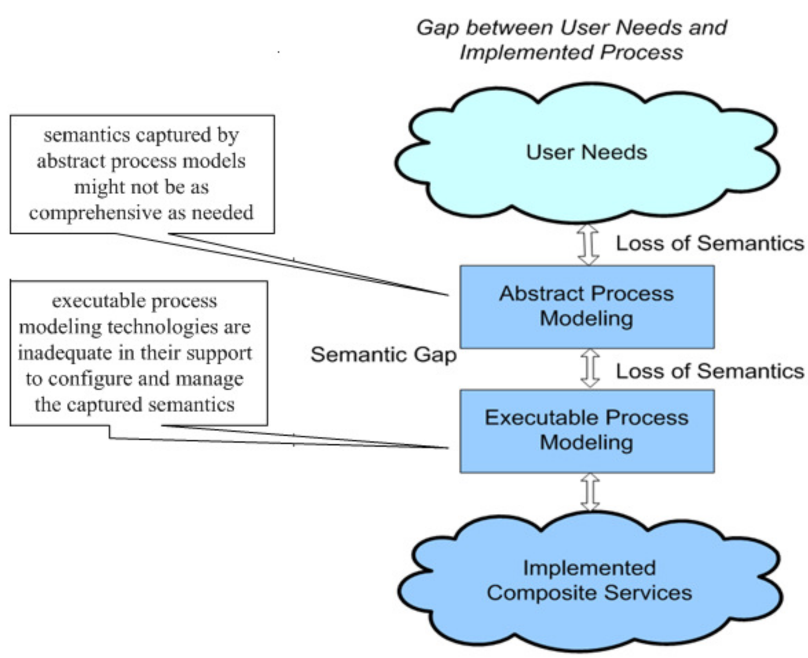


Figure 5. Loss of semantics in process modeling for building composite services

This semantic gap essentially results from the following causes:

i) Lack of a methodology that clearly describes the user processes and their interactions.

ii) Lack of technology that can use a conceptual model of user processes for the purpose of service selection and discovery.

The user processes must be clearly defined by using an appropriate formalism. This formalism will mitigate ambiguity in determining the user processes. A machine-actable conceptual model on the other hand would determine the concepts involved in user processes and use them for the purpose of service selection and discovery. The adverse effects of this semantic gap problem in building composite services are as follows:

i) Increase in the cost of developing composite services.

ii) Increase in the amount of time involved in developing composite services.

iii) Difference between features of the implemented system and user needs.

E. Dissertation Approach

In this dissertation, the approach to solve the semantic gap problem explained in Section C of Chapter 1 involves proposing a framework for the development of composite services. This framework termed the Process-as-a-Concept (PAAC) Framework will mitigate this semantic gap, and will thereby reduce the adverse effects mentioned in Section A. This framework will be based on conceptualizing a process description and converting the conceptual description into its machine-actable representation. Chapter 4 contains a comparison of the PAAC Framework to the existing solutions in reducing this semantic gap problem.

F. Conclusion

In this chapter, an explanation of the problem of semantic gap in software development is provided and an example is included for clarification. The chapter contains an identification of the relationship of this problem to the area of service composition. The adverse effects resulting from this problem are also discussed in this chapter. Last, Chapter 1 provides a very brief overview of the solution that is proposed to mitigate this problem and is presented in the forthcoming chapters in this dissertation. Chapter 2 provides a discussion on various methods and technologies that can be utilized in reducing the semantic gap problem. In Chapter 3, the PAAC Framework for reducing the semantic gap problem in building composite services is discussed. Chapter 4 includes the identification of key contributions of the PAAC Framework, as well as comparison of the PAAC Framework with some of the solutions discussed in Chapter 2.

CHAPTER 2

SURVEY OF METHODS AND TECHNOLOGIES AVAILABLE FOR MITIGATING THE PROBLEM OF SEMANTIC GAP

In this chapter, we survey some of the existing methods and technologies that can be useful in mitigating the problem of semantic gap described in Chapter 1. We analyze the usage of these methods and technologies in mitigating this problem. A comparison among some of these methods and technologies is drawn to estimate their usefulness. Last, the chapter concludes with a discussion of their helpfulness in designing an integrated tool essential for building a software system.

A. Tools Used for Capturing and Representing User Requirements

Requirements management [19] [20] involves establishing and maintaining agreement between customer and developer on both technical and non-technical requirements. The International Council on Systems Engineering (INCOSE) [21] has listed the prominent requirement management tools and the common features provided by these tools. Requirement management tools facilitate structured organization of user requirements and therefore reduce the time and cost involved in developing software systems. Table 1 depicts some of the common features ideally associated with these tools, and Table 2 lists some of the tools.

Table 1

Common features of the requirement management tools

	Common Features of the Requirement Management Tools as listed by INCOSE
1	Determination of requirements/identification
2	Determination of graphical or textual system element structure
3	Configuration management
4	Documents and other output media
5	Interaction with other tools
6	Adherence to standards of INCOSE, IEEE, or other organizations
7	Single/Multiple user environments

Table 2

Some of the software requirement management tools listed by the International Council on Systems Engineering

Product Name	Corporation	Website
Accept Requirements	Accept Software	http://www.acceptsoftware.com
Aligned Elements Version 1.5	Aligned AG	http://www.aligned.ch
Avenqo PEP Version 1.2	Avenqo	http://www.avenqo.com
CASE Spec Version 8.15	Goda Software	http://www.casepec.net/products.htm
Cognition Cockpit Version 5.1	Cognition Corporation	http://www.cognition.us
Contour by Jama Software Version 2.4	Jama Software	http://www.jamasoftware.com
Core Version 5.1.5	Vitech Corporation	http://www.vitechcorp.com
SoftREQ	Software Requirements, Inc.	http://www.softreq.com
Telelogic DOORS Version 9	Telelogic, An IBM company	http://www.telelogic.com/doors

B. Domain Knowledge Representation and Design Tools

In this section, an analysis of a variety of tools used to represent domain knowledge representation and design of a software system is presented.

Newell [22] states that "Knowledge Representation (KR) has long been considered one of the principal elements of Artificial Intelligence, and a critical part of all problem solving." Additionally, Welty [23] states in his dissertation that "the subfields of KR range from the purely philosophical aspects of epistemology to the more practical problems of handling huge amounts of data. This diversity is unified by the central problem of encoding human knowledge - in all its various forms - in such a way that the knowledge can be used." Often, knowledge can be represented with the use of ontology. Methods such as Developing Ontology Grounded Methods and Applications (DOGMA) use the approach of representing domain knowledge in the form of ontology.

Sommerville [24] defined software design as "deriving a solution which satisfies software requirements." Software design tools facilitate the process of designing a software system by providing an environment for it. In this dissertation, we categorize the design tools into two categories:

i) Tools used to provide representations of software design.

ii) Tools that use these representations to provide the actual software design.

In the first category, we identify tools such as UML and SysML. In the second category, we include tools such as Telelogic TAU, MagicDraw, and UML Graph.

B.1 CommonKADS

CommonKADS [25] is a methodology used to support structured knowledge engineering. Over the years, CommonKADS has been gradually developed in the context of the European ESPRIT IT program [26] [27]. CommonKADS provides emphasis on conceptual structure of knowledge of an enterprise system and is mainly used to store this knowledge, which includes tasks, communication, and business processes, by using a knowledge modeling approach.

In the context of CommonKADS, a task is mainly considered a subpart of a business process that contains attributes such as goal orientation, handling of inputs and outputs in a structured and controlled way, resource consumption, and knowledge representation. A task represents the lowest level of detail of a business process. The breakdown of processes into tasks is carried out by considering the attributes of intaking, archiving, decision making, notifying, reporting, paying, and controlling quality [25]. Each task is given task significance on a scale of 1 to 5. CommonKADS classifies tasks into two categories: analytic tasks and synthetic tasks. The analysis of tasks in CommonKADS uses a three-dimensional information model that consists of the following views:

i) Functional view – Consists of the decomposition of the tasks into subtasks and includes their inputs and outputs to provide their information flow network.

ii) Static information structure – Contains a description of information content, and structure of objects that are handled in the task. The UML class diagram is employed to model the information structure.

iii) Control and dynamic view – Provides a description of the temporal order and control over the subtasks and thereby provides information on triggering events, decision-making points, and knowledge related to the aspect of time.

B.2 Concept Maps

Novak and Cañas [28] defined concept maps as "graphical tools for organizing and representing knowledge." These maps include concepts, usually enclosed in circles or boxes of some type, and relationships between concepts; such relationships are indicated by a line connecting two concepts [28]. A concept map allows a computer literate to illustrate an idea, concept, or domain knowledge to be described in a graphical form [29] [5]. Concept maps provide meaningful learning based on psychological learning. They satisfy the conditions of meaningful learning, mainly because they provide conceptually clear presentation of knowledge and thereby, identify large general concepts held by the learner before assimilating instructions on more specific concepts.

The psychological foundation of concept maps as described by Novak and Cañas [28] is as follows: "Learning of concepts is primarily a discovery learning process, where the individual discerns patterns or regularities in events or objects and recognizes these as the same regularities labeled by older persons with words or symbols." Figure 6 depicts a basic form of a concept map. In this figure, we have two concepts, Concept 1 and Concept 2. The relationship between these two concepts is identified by the label associated with the arrow pointing from Concept 1 to Concept 2.

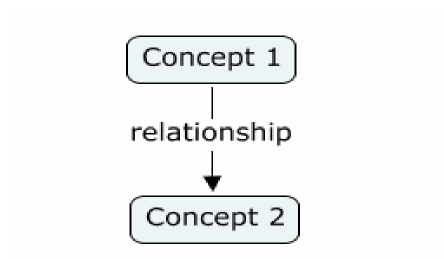


Figure 6. Basic form of a concept map

The interesting aspect of using a concept map tool is that it allows the domain expert to represent the knowledge in a graphical form that can be easily processed by the human mind. Its capability can be harnessed by converting the concept map into an XML-based format. The conceptual information thus obtained in the form of XML acquires the attribute of machine actability and can be used to create semantic data in the form of Resource Description Framework (RDF) and Web Ontology Language (OWL). Concept maps allow a two-level representation of domain knowledge information, as described in Table 3.

Table 3.

Levels of domain representation by concept maps

Level	Description
1	Concept maps allow the domain expert to build a schematic representation of the domain knowledge. This information can be easily comprehended by every individual involved in the development of a software system.
2	The concepts can be converted into machine-actable XML format that can be used by programs capable of parsing and utilizing this information.

Note: From "Representing processes as concepts: Towards reducing semantic gap," by V. P. Gurupur and R.S. Sadasivam, *Proc. 12th SDPS Transdisciplinary Conf. –Workshop on Integrated Systems, Design and Process Science*, Montgomery, AL, 2009. Copyright 2009 by Society for Design Process Science (SDPS). Adapted with permission.

Concept maps have been shown to be understandable to domain experts. Figure 7 depicts a concept map of the airline ticket service process. The models generated with the use of concept maps can then be processed by the human mind as well as generating underlying specifications (e.g. in XML format) that can then be transformed to an executable format.

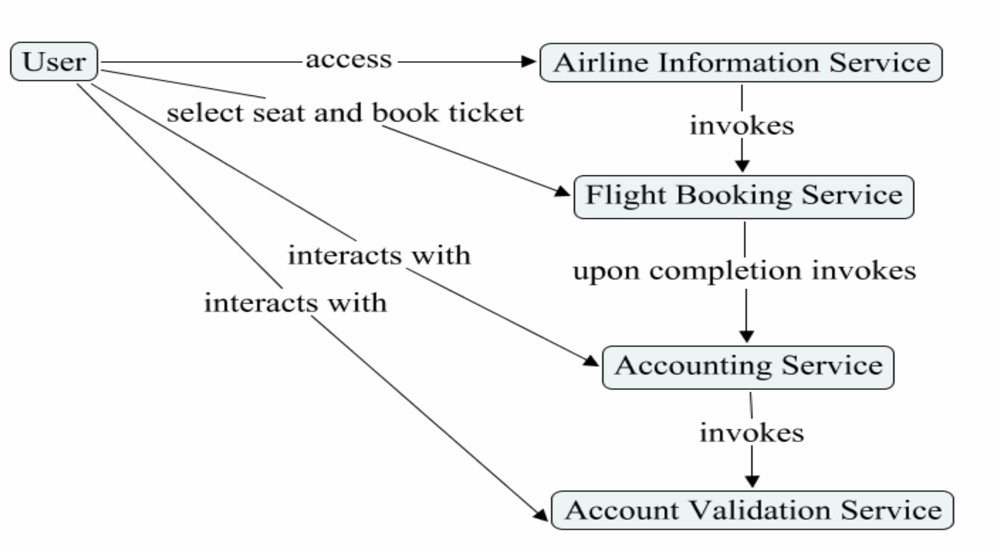


Figure 7. Concept map depiction of example airline ticket service process

Note: From "Representing processes as concepts: Towards reducing semantic gap," by V. P. Gurupur and R.S. Sadasivam, *Proc. 12th SDPS Transdisciplinary Conf. –Workshop on Integrated Systems, Design and Process Science*, Montgomery, AL, 2009. Copyright 2009 by Society for Design Process Science (SDPS). Adapted with permission.

Because the concept map provides an efficacious tool for determining user semantics, the loss of semantics at this stage can potentially be minimized. Second, the concept map model can then be transformed to an executable format in an automated fashion; this capability can further reduce the loss of semantics in process modeling. If the executable process modeling technology is chosen carefully, the loss of semantics because of this transformation can also be kept to a minimum. However, one of the drawbacks of concept maps is that they have a limited ability to represent causal relationships and therefore may have limited use in modeling complex processes. Table 4 depicts the list of software tools available to build concept maps.

Table 4.

Software tools available for building concept maps

	Software	Platform
1	Axon Idea Processor	Windows
2	3D Topicscape	Windows
3	Inspiration	Macintosh and Windows
4	Kidspiration	Macintosh and Windows
5	MindManager	Macintosh and Windows
6	MindMapper	Windows
7	MindView	Macintosh and Windows
8	CMapTools	Linux, Macintosh and Windows

B.3 Mind Maps

Mind maps are diagrammatic representations used to explore ideas; to enhance work morale or problem solving, team building, or synergy creating activity; and to represent a business strategy or anything that can be visualized by the human mind. In his book, *Mind Maps at Work*, Tony Buzan [30] defined mind map as "a colorful visual form of note-taking that can be worked on by one person or a team of people." Mind maps existed as early as 300 CE, when Porphyry of Tyre, a Neoplatonic philosopher, used a graphical visualization of the concept categories of Aristotle [31].

B.4 SysML

SysML, a modeling language for representing enterprise systems was developed by Object Management Group (OMG) [32] [27]. OMG is an international, open membership, not-for-profit computer industry consortium. SysML supports a broad range of systems such as hardware, software, processes, information, personnel, and system-of-systems [33]. As stated by OMG, SysML represents a subset of UML 2.0 with extensions needed to satisfy the requirements of UML™ for systems engineering. SysML provides several advantages over traditional UML [34] for representing software systems [35]. While UML is software centric, SysML is centered on the systems engineering. The main reason for the development of SysML is to reduce the document intensive nature of systems engineering processes. Table 5 lists the tools available for building SysML models [35].

Table 5.

Software tools available for building SysML models

	Tool	**Organization**
1	SysML Toolkit	EmbeddedPlus
2	Rhapsody	I-Logix
3	Enterprise Architect	Sparx
4	ARTiSAN Studio	Artisan Software
5	TAU Generation2	Telelogic
6	Visio	Microsoft

B.5 Software Factory

Greenfield and co-authors [36] define a software factory as a "software product line that configures extensible tools, processes, and content using a software factory template based on a software factory schema to automate the development and maintenance of variants of an archetypical product by adapting, assembling, and configuring framework-based components." The software factory template essentially includes both code and metadata whereas a software factory schema includes concerns and abstractions that may be represented in the form of grids.

The central idea of software factories is to automate the process of software development. To achieve this goal, the software requirements are first given a structured form and then provided to a code automation system. The philosophy behind the development of software factories provides emphasizes on determining abstract information and using it to implement working software systems.

B.6 Task System Model

A task system model (Figure 8) is defined by Mills and Tanik [37] as "a process engineering formalism that supports the development and maintenance of distributed process systems." The task system model was initially designed to build operating systems. The primary aim of this model was to facilitate the tasks' cooperating toward a common goal to be executed in parallel. In the context of operating systems, the task system model provided the attributes of speed independency and determinacy to the tasks involved in it [38]. To achieve determinacy, the physical systems on which the task systems were executed were represented by an ordered set. Because of the factor of determinacy, the task system models proved useful in preventing deadlocks during the functioning of an operating system.

Mills and Tanik [37] provided a path for using task system models for an ordinary process such as a car wash. This path extended the idea of the task system model by including the factor of non-determinacy. Some of the main features of a task system model include:

i) Increases timeliness of a process system through increased parallelism and maximizing the utilization of the resources.

ii) Involves a set of tasks, depicted by a precedence graph.

iii) Each task is associated with a resource or a set of resources.

iv) Acquisition and assignment of resources.

v) Execution sequence of the tasks indicated by a directed graph termed as a precedence graph.

A task system model involves two types of tasks:

i) Sequential task: A task that requires synchronization.

ii) Independent task: A task that is independent and can be executed in parallel.

A typical task system model involving processes A, B, C, D, E, and F can be denoted by

$C = (T, <^{*})$ where (1)

$T = \{ A, B, C, D, E, F \}$ (2)

$C_{ABCDEF} = \{(A, B), (A, C), (B, D), (C, E), (D, F), (E, F)\}.$ (3)

Figure 8 depicts a precedence graph of this task system model. If we assume that this task system involves five resource types, R1, R2, R3, R4, and R5, the set of resource types, $\rho_{ABCDEF,}$ can be denoted as

$\rho_{ABCDEF} = \{ RT_{R1,} RT_{R2,} RT_{R3,} RT_{R4,} RT_{R5} \}.$ (4)

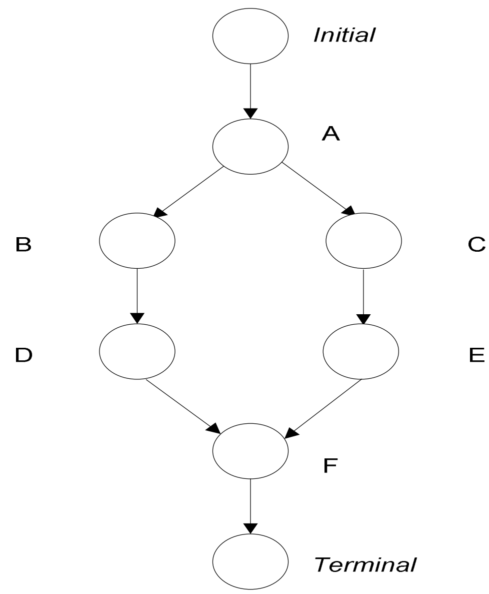


Figure 8. Precedence graph of the task system model C_{ABCDEF}

C. Tools and Languages Associated with Service Oriented Architecture and Semantic Web

Service Oriented Architecture was defined by Newcomer and Lomow [39] as "a style of design that guides all aspects of creating and using business services throughout the development life cycle" and W3C [40] define a web service as "a software system designed to support interoperable machine-to-machine interaction over a network." Service Oriented Architecture relies on publishing, distributing, and utilizing a web service to build a distributed computing system. Individual web services, sometimes termed services, can be composed and integrated to form a composite service by using the process of service composition. Dustar [17] defined service composition as, "those activities required to combine and link existing web services (atomic and composite services) and other components to create new processes." Semantically related services can be composed by using the process of semantic service composition. W3C defined semantics of a service as “the behavior expected when interacting with the service.”

W3C [40] define semantic web as “a common framework that allows data to be shared and reused across application, enterprise, and community boundaries.” Semantic Web is an extension of the World Wide Web in which information is given well-defined meaning and thus better enables computers and people to work together [40]. The semantic web is the artificial intelligence part of the World Wide Web.

C.1 Resource Description Framework

Daconta, Obrst, and Smith [41] define RDF as “an XML-based language to describe resources.” The most elementary form of information in RDF is an RDF triple (Figure 9) that consists of a subject, a predicate, and an object. Consider an example, “David is the author of book 1.” Here *David* is the subject, *author* happens to be the predicate, and *book 1* is the object. RDF has been used to develop several applications, such as RDFPic [42], and RDF is most commonly used to build software applications that handle metadata. Table 6 lists some of the RDF relevant tools listed by W3C.

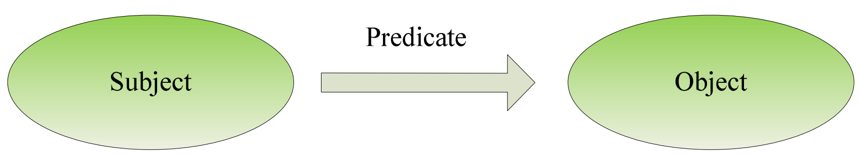


Figure 9. Resource Description Framework triple

Table 6.

List of the Resource Description Framework tools listed by World Wide Web Consortium

	Tool Name	Website
1	3Store	http://www.aktors.org/technologies/3store/
2	4Suite	http://4suite.org/index.xhtml
3	ARC RDF Store	http://arc.semsol.org/
4	ActiveRDF	http://activerdf.org/
5	Allegro Graph RDF Store	http://www.franz.com/agraph/allegrograph/
6	CARA	http://cara.sourceforge.net/
7	D2R Server	http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/
8	Disco	http://www4.wiwiss.fu-berlin.de/bizer/ng4j/disco/

C.2 Web Ontology Language

According to W3C, the OWL was designed for use by applications that need to process the content of information instead of just presenting information to humans [43]. OWL is an XML-based language for ontological representation of information. In comparison with RDF, OWL is a stronger language because OWL provides more machine interpretability. It is commonly used to store metadata for medical applications [44]. Table 7 lists some of the tools for using OWL. OWL is composed of three sublanguages [43]:

i) OWL Lite – Uses a classification hierarchy with simple constraints.

ii) OWL DL – Provides the user with maximum expressiveness while retaining computational completeness.

iii) OWL Full – Provides the user with maximum expressiveness and syntactic freedom, with no computational guarantees.

Table 7.

List of tools for using Web Ontology Language

	Editor Tool	Website
1	protégé	http://protege.stanford.edu/
2	Triple20	http://www.swi-prolog.org/web/Triple20.html
3	RacerPro	http://www.franz.com/agraph/racer/

C.3 Business Process Execution Language for Web Services

IBM [45] defined the Business Process Execution Language for web services (BPEL4WS) as “a formal specification for business processes and business interaction protocols.” It is mainly used to build composite web services. BPEL4WS is used to define two processes:

i) Abstract process – Used to define the business protocol of an enterprise system.

ii) Executable process – Used to define the nature and sequence of web service interactions.

C.4 Unified Medical Language System

The National Institutes of Health (NIH) [46] define Unified Medical Language System (UMLS) as “a language system to facilitate the development of computer systems that behave as if they understand the meaning of the language of biomedicine and health." By using UMLS, NIH has facilitated the development of tools and databases (knowledge source) that can perform the required operations on health data and information associated with it. UMLS knowledge is distributed across three sources:

i) Metathesaurus – A database consisting of the vocabularies and their concepts. The source vocabularies in the metathesaurus are organized as a single, fully specified data source. The data source is organized in the form of concepts.

ii) Semantic Network – Provides a categorization of concepts mentioned in the Metathesaurus and provides the relationships between them.

iii) SPECIALIST Lexicon and Lexical Programs – Contains required lexical information for SPECIALIST Natural Language Processing System. This source is basically in place to facilitate the natural language processing of the information contained in the data sets.

C.5 DBpedia

DBpedia is a data set that is available in the form of RDF triples [47]. The data is extracted on a real time basis from Wikipedia. Therefore, DBpedia is a structured form of the contents available in Wikipedia. Because Wikipedia is a growing data set of information available to the Internet users, DBpedia can be perceived as a sophisticated, ever growing data set that dwells on the success of Wikipedia. The advantage of structuring the information available in Wikipedia is that this information can be queried by using semantic web tools such as SPARQL. The query outputs the requested meta information in the form of RDF. Some of the key features of DBpedia are as follows:

i) The data are dumped from recent Wikipedia dumps.

ii) The data set facilitates 91 different languages.

iii) The data set is associated with a SPARQL endpoint for querying data.

D. Frameworks Involved in Building Enterprise Systems

Zachmann [48] defined a framework as "simply a logical structure for classifying and organizing the descriptive representations of an enterprise that are significant to the management of the enterprise as well as to the development of the Enterprise's systems." This explication could be perceived as a general definition of a framework for enterprise systems. In the case of the Federal Enterprise Architecture Framework [49], frameworks are defined as "logical structures for classifying and organizing complex enterprise architecture information."

Architecture frameworks such as the Artificial Intelligence Design Framework [27], developed by Tanik, and the Composite Process-Personalization Framework [18], developed by Sadasivam, provide emphasis on the general architecture involved in developing software systems. In this dissertation, we develop an architecture framework for bridging the semantic gap in the process of service composition. The next subsections contain discussions of other frameworks, including DOGMA, which uses information available in the form of ontology to build enterprise software systems.

D.1 DOGMA Ontology-Based Framework

DOGMA is an ontology-based framework currently under development at Vrije Universiteit Brussel. A primary motivation for this project is to build machine-actable semantic resources that can be shared among computer scientists and software developers to build computer science applications. Mustafa Jarrar and Robert Meersman [50] stated that “by sharing an ontology, autonomous and distributed applications can meaningfully communicate to exchange data and thus make transactions interoperate independently of their internal technologies.” This framework uses ontology representation at three different levels:

i) Domain level – Representation of ontology pertaining to a specific domain that can provide general information about a domain.

ii) Application kind – Use of ontology to represent a set of applications that fall under the same category in a given domain.

iii) Specific application – Involves representation of information adhering to a specific application in the form of an ontology.

D.2 Department of Defense Architecture Framework

The Department of Defense Architecture Framework (DoDAF) [51] provides the necessary information for representing, developing, and understanding defense architectures. The framework addresses the following audiences:

i) Decision maker – Helps the decision maker understand the application of architecture and thus facilitates the process of decision making.

ii) Architect – Helps the architect comprehend the data-centric approach toward the development of architecture.

iii) Manager – Helps the manager understand the aspects of data management and maintenance.

The process of analysis of the architectures developed by using this framework involves data modeling and visualization. Here we have three levels of data models used:

i) Conceptual level – Involves representation of concepts by using terms that are familiar to the user.

ii) Logical level – Provides a more formal representation of the model by using data representation consisting of semantic and domain level information.

iii) Physical level – Contains all the necessary information to implement data-bases.

D.3 Bio2RDF

Bio2RDF is a bioinformatics knowledge system built with the use of semantic web technology [52] [53]. The knowledge system accepts information represented with the use of a text file or an XML-based document. The Bio2RDF system can be queried by using an online query system, which accepts queries written by using SPARQL, and displays the result in an XML-based format. The system currently includes at least 46 million RDF documents. This system is built with the use of the JSP rdfizer, the Sesame open source triplestore technology, and OWL ontology. The Rdfizer, a program that converts the existing data into an RDF representation, adheres to an underlying philosophy that all existing knowledge can be represented with the use of the RDF representation. A broad range of file types can be converted into an RDF representation by Rdfizers like SIMILE [54].

E. Conclusion

This chapter discusses tools, languages, frameworks, methods, and formalisms that can be used to mitigate the communication gap at various levels of developing a software system. No one of them does sufficiently bridge the semantic gap discussed in Chapter 1. However, in this dissertation, an attempt to use a combination of some of these techniques to bridge the semantic gap is presented.

CHAPTER 3

PROCESS-AS-A-CONCEPT FRAMEWORK

A. Overview of the Approach

In Chapter 1, we established the need for bridging the semantic gap in the context of building composite services. The dissertation then described in Chapter 2 the methods, frameworks, and tools that can be used in bridging this gap. At the end of Chapter 2, we concluded that, to build a framework we would need a combination of the some of the tools and methods discussed in the chapter. In Chapter 3, the dissertation describes the PAAC Framework mentioned at the end of Chapter 1. This framework is intended to bridge the semantic gap between the user needs and the process of service selection and discovery (Figure 10).

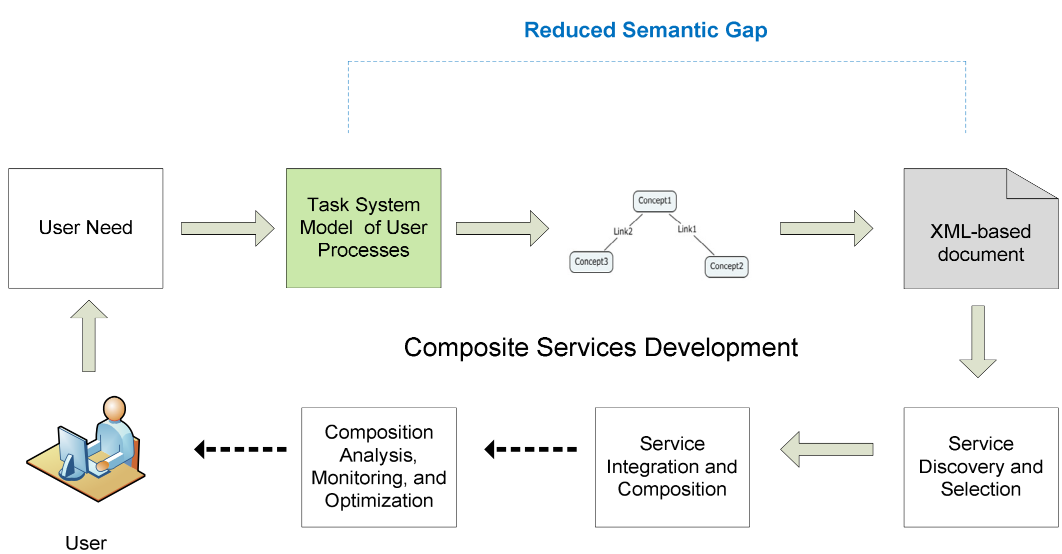


Figure 10. Overview of the Process-As-A-Concept Framework

Figure 10 provides an overview of the PAAC Framework and its significance in solving the problem of semantic gap. The framework involves the following set of operations in building composite services:

i) Identifying the user needs and then the user processes from them.

ii) Developing a task system model representing the user processes and their interactions.

iii) Using the task system model to develop a concept map.

iv) Building composite services by using the XML-based ontology document built with the use of concept maps.

v) Identifying and selecting the appropriate services.

vi) Integrating and composing the services to form a composite service.

vii) Performing an analysis of the composite service and optimizing the service if required.

A.1 Five Main Reasons for Using Concept Maps

The main reasons for using concept maps are as follows:

i) Concept maps provide meaningful learning because they offer conceptually clear representation of knowledge [28] [29].

ii) Concept maps are used to provide a strong foundation for the development of domain specific ontology [28] [29].

iii) Concept maps allow thorough determination of both syntactic and semantic elements of processes of composite services.

iv) Concept maps allow domain experts and developers to work together to design abstract models of the user domain [28].

v) A concept map can be converted into an executable format and thereby reduces the loss of semantics in process modeling.

A.2 Three Main Reasons for Using Ontology to Describe User Processes

The three main reasons for using ontology to describe user processes are as follows:

i) Ontology provides a machine-actable form of semantic information contained in the concept map [28] [29].

ii) An XML-based ontology description can be easily parsed by using a suitable parser to extract information on user processes.

iii) Information depicted in the form of an ontology reduces the ambiguity involved in a natural language representation.

A.3 Identifying User Processes from User Needs

Identifying user processes from user needs is a critical step in developing composite services. This step requires a good interaction between the user and the developer. Deciphering the user processes from user needs involves identifying the following elements of user needs:

i) The processes involved in building a solution for the user needs.

ii) The interaction among user processes.

iii) The flow of control among user processes.

A.4 Representing User Processes as Concepts

The most critical step in the PAAC Framework consists of representing the user processes as concepts. Representation of user processes as concepts involves the following aspects:

i) Developing a task system model representing the user processes and their interactions.

ii) Using the task system model to develop a concept map.

A.5 Building Composite Services by Using Concept Maps

Because concept maps can be easily converted into a machine-actable, XML-based ontology document, these maps play a vital role in building composite services. Using concept maps to build composite services mainly involves the following steps:

i) Converting concept maps into an XML-based document such as an OWL document.

ii) Extracting the information on user processes from the XML-based document by using a suitable parser.

iii) Providing the extracted information to the service discovery and selection process.

B. Converting a Task System Model into a Concept Map

A methodology with which to convert a task system model into a concept map involves the following steps:

i) Identify the resources involved in the task system model.

ii) Identify the interaction between the resources, and build a table of RDF triples indicating the resources and their interactions.

iii) Build a concept map for every RDF triple in the table, and use these maps for the task system model.

In this dissertation, we discuss this methodology by using two different case studies: a) a car wash example and b) a gene linkage application.

B.1 Analyzing the Conversion of a Task System Model into a Concept Map by Using a Car Wash Example

In this dissertation, we first analyze the conversion of a task system model into a concept map; here we use the car wash example employed by Mills and Tanik [37]. We then analyze the means of carrying out this conversion in the case of a gene linkage application. Table 8 describes the resources involved in the task system model. The first column provides the names of the resources, and the second column describes the relevance of each resource in terms of the car wash example.

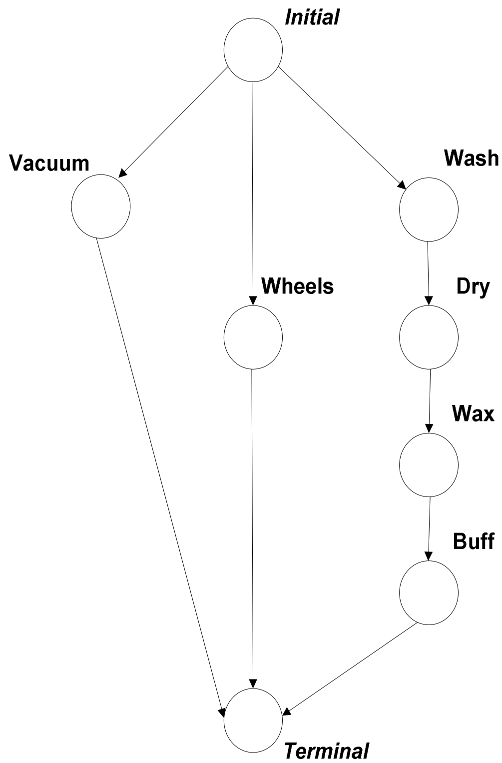
Figure 11 provides the precedence graph of the car wash example [37]. This graph contains the initial and final states per convention, as well as the intermediate states involved, in the car wash example. The task system model for the car wash example considered here is converted into a concept map by using the steps identified in the beginning of section B.

Table 8.

Description of resource types for the car wash example

Resource Type	Description
Staff	A group of individuals who use other resources for cleaning the car
Vacuum	A device used to clean the interior of the car
Soap	Foaming liquid used to clean the exterior of the car
Sponges	A paraphernalia used with the soap to clean the car
Towels	A piece of cloth used to dry the exterior of the car
Wax	Used to shine the exterior of the car
Rags	Used to dry the wax
Brushes	A paraphernalia used to clean the wheels
Wheels	A part of the exterior of the car
Water Hose	A device used to rinse the car to remove soap
Exterior	External part of the car
Interior	Internal part of the car

Table 9 depicts the RDF triples built by using the interactions among the resource types. We use the RDF triples from every row to build a concept map of the system. Let us consider the first RDF triple, Staff -> uses -> Vacuum; the corresponding concept map would be as indicated in Figure 12. Aggregation of the second RDF triple, Vacuum -> cleans -> Interior; gives us the concept map depicted in Figure 13. First, a concept map is built by using the RDF triples represented in Table 9. Next, these individual concept maps are linked by using the information depicted in Table 10.



C_{Carwash} = {(*Initial*, Vacuum), (*Initial*, Wash), (*Initial*, Wheels), (Wash, Dry), (Dry, Wax), (Wax, Buff), (Vacuum, *Terminal*), (Buff, *Terminal*), (Wheels, *Terminal*)}

$\rho_{\text{Carwash}} \equiv \{ RT_{\text{Staff}}, RT_{\text{Vacuum}}, RT_{\text{Soap}}, RT_{\text{Sponges}}, RT_{\text{Towels}}, RT_{\text{Wax}}, RT_{\text{Rags}}, RT_{\text{Brushes}}, , RT_{\text{Interrior}}, RT_{\text{Exterior}}, RT_{\text{Wheels}}, RT_{\text{Water Hose}} \}$

Figure 11. Precedence graph of the task system model for the car wash example

Table 9.

Resource Description Framework triples built by using resource dependency for the car wash example

Triple	Subject	Predicate	Object
Triple 1	Staff	uses	Vacuum
Triple 2	Vacuum	cleans	Interior
Triple 3	Staff	uses	Soap
Triple 4	Soap	cleans	Exterior
Triple 5	Sponges	work_with	Soap
Triple 6	Staff	uses	Wax
Triple 7	Wax	shines	Exterior
Triple 8	Water hose	removes	Soap
Triple 9	Towels	dry	Exterior
Triple 10	Water hose	cleans	Wheels
Triple 11	Rags	dry	Wax
Triple 12	Brushes	cleans	Wheels

Figure 14 depicts the concept map built from all RDF triples that are represented in Table 10 and are based on the information given in Table 9. In this way, a methodology for converting a task system model into a concept map is provided.

Table 10.

Linking Resource Description Framework triples for the car wash example

Linking RDF Triple 1	Linking RDF Triple 2	Linking Resource
Triple 1	Triple 2	Vacuum
Triple 3	Triple 1	Staff
Triple 4	Triple 3	Soap
Triple 5	Triple 4	Soap
Triple 8	Triple 10	Water hose
Triple 11	Triple 6	Wax
Triple 6	Triple 7	Wax
Triple 9	Triple 7	Exterior
Triple 8	Triple 5	Soap
Triple 12	Triple 10	Wheels

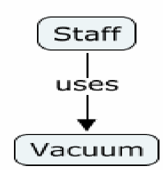


Figure 12. Concept map built from the first Resource Description Framework triple

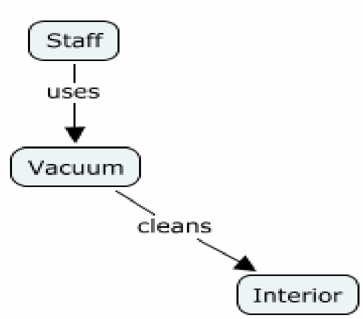


Figure 13. Linking concept maps by using Table 3

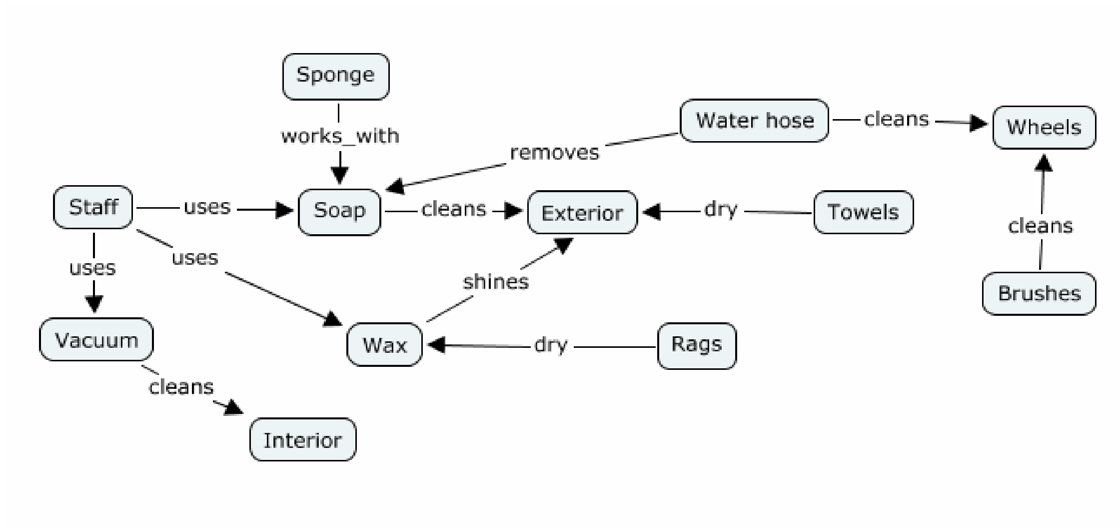


Figure 14. Concept map built from all Resource Description Framework triples

B.2 Validation of Converting a Task System Model into Concept Maps

The validation of the aforementioned methodology of converting a task system model into a concept map is carried out by using this methodology to build software applications. In this dissertation, an example of using this methodology to build a gene linkage application is provided.

C. Building a Gene Linkage Application by using the PAAC Framework

C.1 Analyzing the Processes Used in Building Gene Linkage Information

Figure 15 depicts the manual process used in linking gene information at the Department of Epidemiology at UAB [55]. This process is both labor intensive and time consuming. In this process, the research assistant manually searches a keyword in a set of articles listed in the NCBI portal [56]. The articles that contain this keyword are then researched for extracting the relevant gene information associated with the article. Because this process is labor intensive, a semi-automated process was built by using the CP^2 Framework [18].

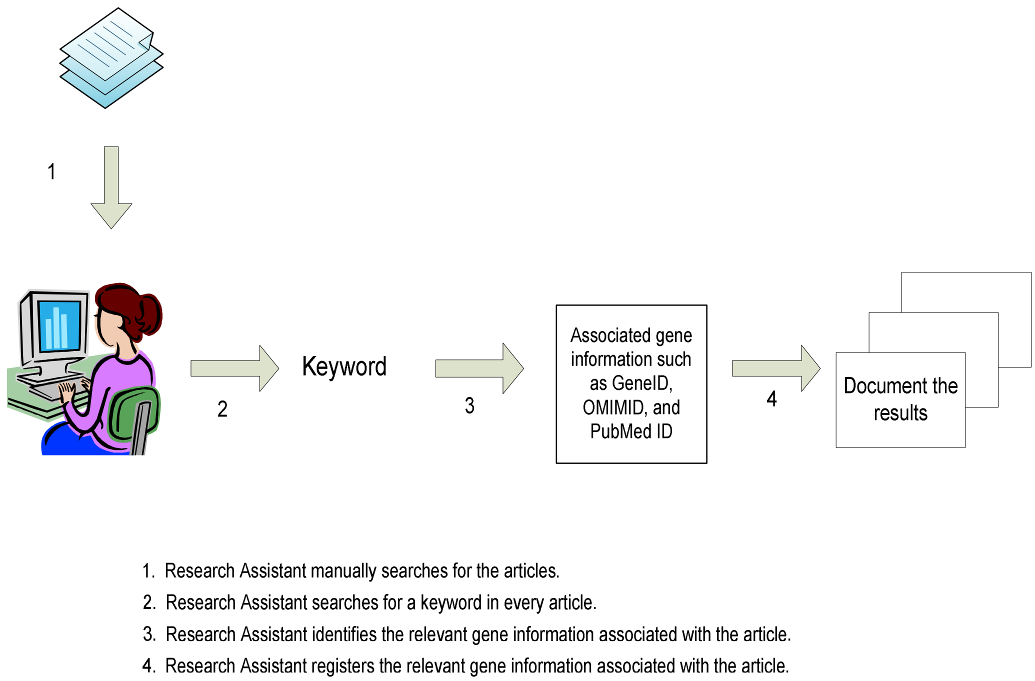


Figure 15. Manual process involved in identifying gene linkage

This semi-automated process extracts the relevant gene information by allowing a software application to accept a parameter from the user, such as a keyword or a chromosome region. Although this process reduces the labor intensiveness of the manual process, it is associated with the following drawbacks:

- The user is expected to know the semantic intricacies of the data and of the terminologies involved in gene linkage.
- The user is also expected to possess the knowledge of the terminologies involved in gene linkage.

C.2 Building the Gene Linkage Process by Using the Process-As-A-Concept Framework

Figure 16 depicts the process of using the PAAC Framework to build the research application. The following steps are taken to build the service application:

i) The domain expert and the software developers build the task system model for the service application.

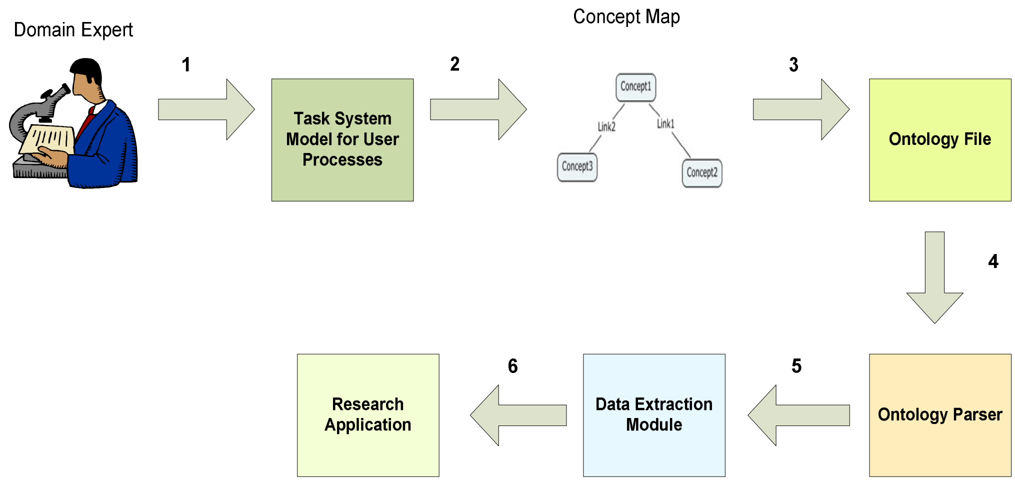
ii) The task system model is used to build the concept map.

iii) The concept map thus obtained is converted into an OWL document.

iv) This ontology file is parsed by a suitable parser to extract the relevant data elements.

v) The extracted information is then provided to the research application

Figure 17 provides the relationship between the methodology and the PAAC Framework and Table 11 provides the resource description for gene linkage application. This information depicted in Table 11 is useful in building the relationships between the resources. For instance, the developer can perceive that both Gene Name and Gene ID are the attributes used to identify a gene. Figure 18 provides the task system model of the gene linkage application. Table 12 depicts the RDF triples built by using the resource description of the resources involved in the Gene Linkage Application. These individual RDF triples are then linked together by using the common linking resource (Table 13).



1. The domain expert identifies the user processes, and this knowledge is used to build the task system model.
2. The concept maps are built by the domain expert.
3. The concept maps are translated into a suitable ontology file such as OWL.
4. The ontology file is parsed by a suitable ontology parser.
5. The ontology parser provides the required information to the Data Extraction Module.
6. The Research Application uses the information gathered by the Data Extraction Module.

Figure 16. Building the gene linkage application using the Process-As-A-Concept Framework

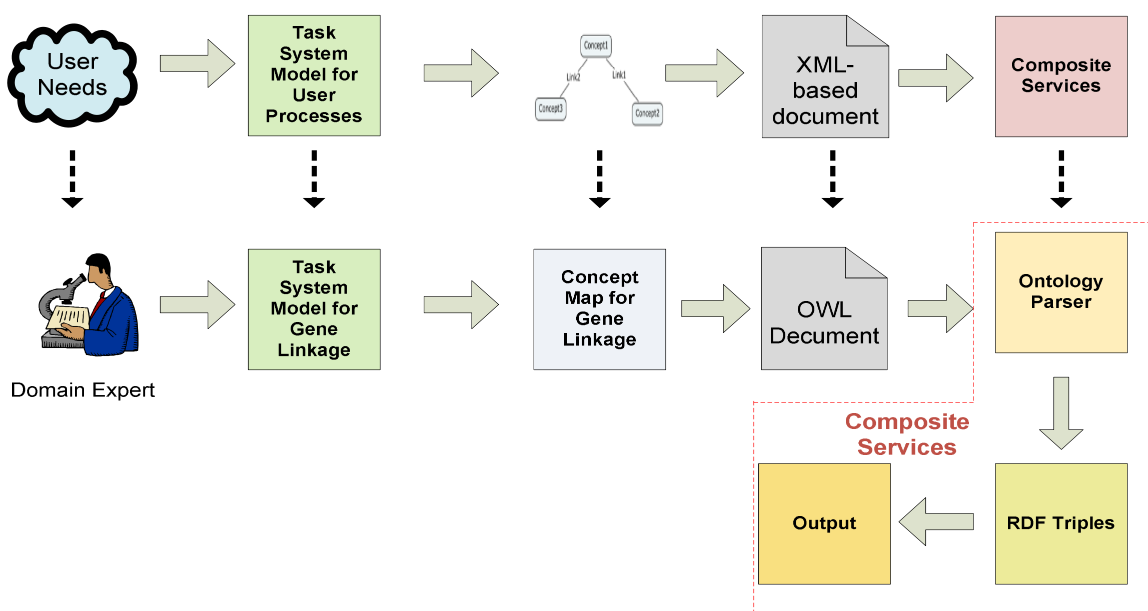
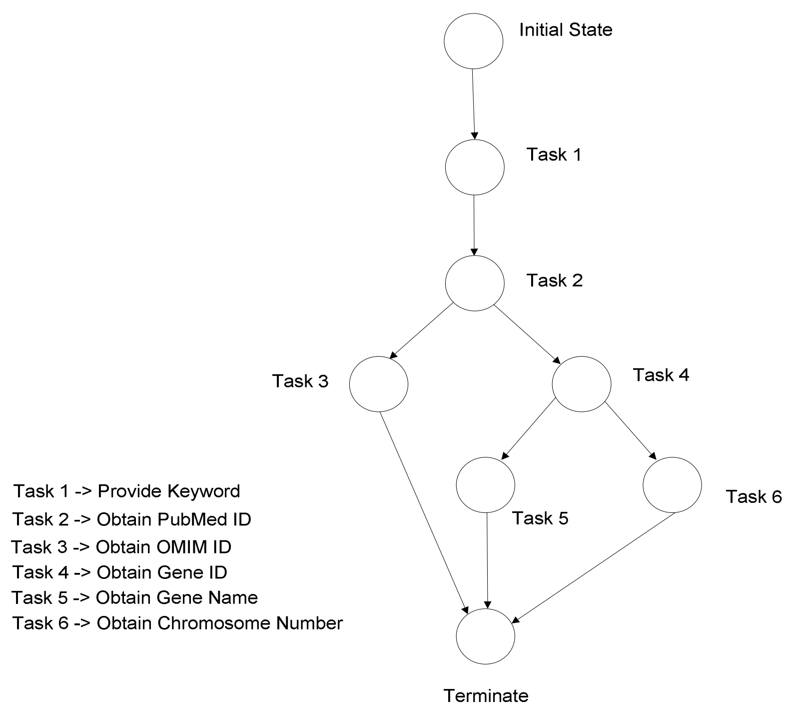


Figure 17. The relationship of the application with the Process-As-A-Concept Framework

Table 11.

Resource description for the gene linkage application

Resource Type	Resource Description
Keyword	A string found in the abstract of a publication listed by NCBI databse
PubMed ID	Identification number provided to every publication listed by NCBI database
OMIM ID	Identification number given to a disease related to human genes and genetic disorders
Gene ID	Identification number provided to every gene listed by NCBI
Gene Name	Name provided to every gene listed by NCBI
Chromosome Number	Identifies the chromosome region



$$C_{GL} = \{(\text{Task 1, Task 2}), (\text{Task 2, Task 3}), (\text{Task 2, Task 4}), (\text{Task 4, Task 5}), (\text{Task 4, Task 6})\}$$

$$\rho_{GL} = \{ RT_{\text{Keyword}}, RT_{\text{PubMed ID}}, RT_{\text{OMIM ID}}, RT_{\text{Gene ID}}, RT_{\text{Gene Name}}, RT_{\text{Chromosome Number}} \}$$

Figure 18. Task system model for the gene linkage application

Table 12.

Resource Description Framework triples built by using resource dependency for the gene linkage application

RDF Triple	Subject	Predicate	Object
Triple 1	Keyword	connected_with	PubMedID
Triple 2	PubMedID	connected_with	GeneID
Triple 3	PubMedID	connected_with	OMIMID
Triple 4	GeneName	connected_with	Chromosome Number
Triple 5	GeneName	belongs_to	GeneID

Table 13.

Linking Resource Description Framework triples using a common resource for the gene linkage application

Linking RDF Triple 1	Linking RDF Triple 2	Linking Resource
Triple 1	Triple 2	PubMedID
Triple 2	Triple 3	PubMedID
Triple 1	Triple 3	PubMedID
Triple 4	Triple 5	GeneName
Triple 5	Triple 2	GeneID

Figure 19 illustrates the concept map derived from Table 13, and Figure 20 depicts the aggregation of the concept maps that was built using individual RDF triples. For instance, Triple 1 and Triple 2 are connected by the common resource PubMedID [57]. By converting a task system model into a concept map, the framework provides a methodology with which a formalism that belongs to the domain of process engineering is converted into a representation that belongs to the area of artificial intelligence.

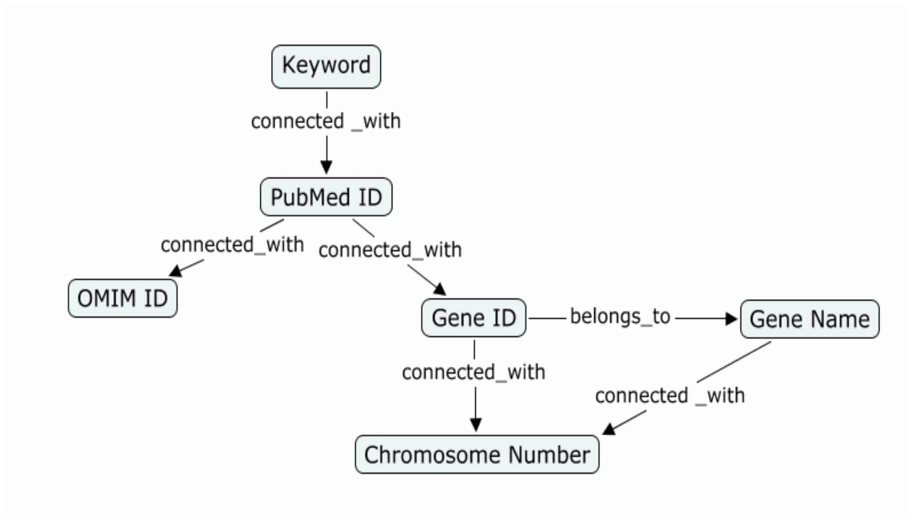


Figure 19. Concept map of the gene linkage application

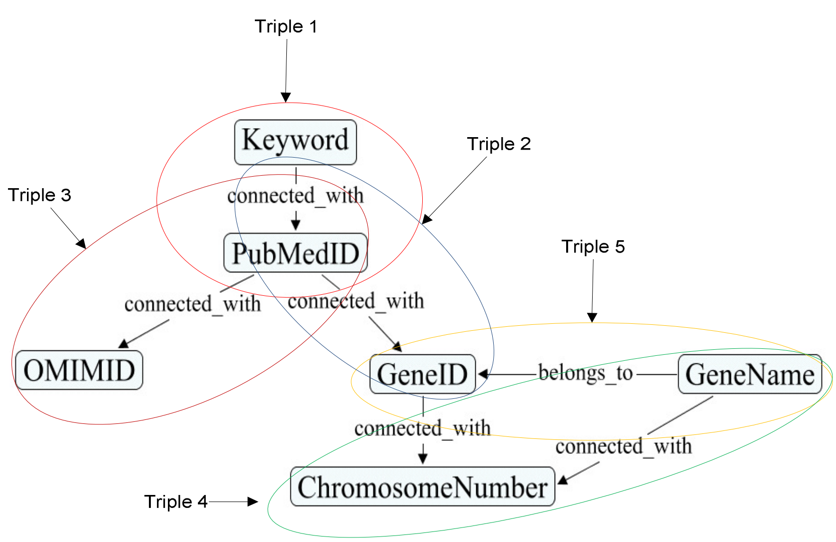


Figure 20. Concept map illustrating the gene linkage application and identifying the Resource Description Framework triples

The representation of a process in the form of a concept is a critical step in building machine-actable representation of domain information. In the next section, we describe the conversion of concept maps into an OWL document and discuss the usage of the document in building an appropriate service.

D. Building an Application Service by Using Concept Maps

In Section C, a discussion on the conversion of a task system model into a concept map was presented. This section provides the details of using a concept map thus obtained to build a service.

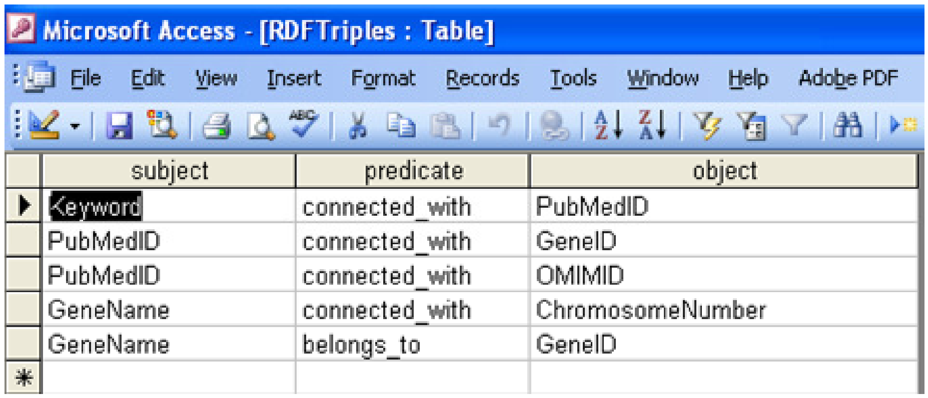
A critical objective of this dissertation is to facilitate the development of abstract information on a required service and then convert that information into executable information. The development of a concept map from a task system model facilitates the development of abstract information needed by a team of users and developers involved in gathering the requirements of a particular software system. Conversion of the requirements thus gathered into machine-actable representation of information provides the required path to build the necessary services for building a composite service.

In the concept map represented in Figures 19 and 20, each object such as Keyword, Gene ID, OMIM ID [58], Gene Name, and Chromosome Number denoted by different concepts can be linked to a whole new concept map resembling a network. This concept map is then converted into the OWL document depicted in Figure 21. With the use of a Java SAX parser [59], the developer can extract from this OWL document the information necessary to construct RDF triples stored in a database. The triples provide a semantic description of the available gene linkage data to a database application built by using Java or .NET technology [60]. The database application built by using Java Server Pages [61] extracts the semantic information stored in the form of RDF triples. This database application extracts the data stored on individual genes and their associated publications by using the semantics described by the RDF triples to provide the output, as indicated in Figure 22. The RDF triples shown in Figure 22 provide the developer the information for building the semantics of the data associated with the gene linkage application. While the actual data is stored in a separate database, the semantics of the actual data is derived by using the concept map built by the domain expert.

```
<rdf:RDF
    xmlns="http://localhost/default#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:daml="http://www.daml.org/2001/03/daml+oil#">
  <owl:Class rdf:about="http://localhost/default#GeneID">
    <connected_with>
      <owl:Class rdf:about="http://localhost/default#ChromosomeNumber"/>
    </connected_with>
  </owl:Class>
  <owl:Class rdf:about="http://localhost/default#GeneName">
    <connected_with rdf:resource="http://localhost/default#ChromosomeNumber"/>
    <belongs_to rdf:resource="http://localhost/default#GeneID"/>
  </owl:Class>
  <owl:Class rdf:about="http://localhost/default#OMIMID"/>
  <owl:Class rdf:about="http://localhost/default#PubMedID">
    <connected_with rdf:resource="http://localhost/default#GeneID"/>
    <connected_with rdf:resource="http://localhost/default#OMIMID"/>
  </owl:Class>
  <owl:Class rdf:about="http://localhost/default#Keyword">
    <connected_with rdf:resource="http://localhost/default#PubMedID"/>
  </owl:Class>
</rdf:RDF>
```

Figure 21. Ontology corresponding to the gene linkage concept map

The application developed to extract the necessary gene data expects the user to provide a keyword. The application then searches for the keyword in the database. Once the keyword is found, the application refers to the table represented in Figure 22 and identifies the necessary attributes related to the keyword to be extracted from the database. These attributes are then extracted from the database; a semantic representation of these data elements, along with the actual data, is provided to the user (Figure 23). Henceforth, the user receives the actual gene data and the relationships between the extracted data (Figure 24).



subject	predicate	object
Keyword	connected_with	PubMedID
PubMedID	connected_with	GeneID
PubMedID	connected_with	OMIMID
GeneName	connected_with	ChromosomeNumber
GeneName	belongs_to	GeneID

Figure 22. Resource Description Framework triples extracted from the ontology depicted in Figure 13

For the Keyword: rat

rat connected_with PubMedID: 15489334
PubMedID: 15489334 connected_with GeneID: 619491
PubMedID: 15489334 connected_with OMIMID: 601665
U58 connected_with ChromosomeNumber: 18
U58 belongs_to GeneID: 619491

rat connected_with PubMedID: 12477932
PubMedID: 12477932 connected_with GeneID: 100129143
PubMedID: 12477932 connected_with OMIMID: 601665
LOC100129143 connected_with ChromosomeNumber: 18
LOC100129143 belongs_to GeneID: 100129143

rat connected_with PubMedID: 16380380
PubMedID: 16380380 connected_with GeneID: 100129143
PubMedID: 16380380 connected_with OMIMID: 601665
LOC100129143 connected_with ChromosomeNumber: 18
LOC100129143 belongs_to GeneID: 100129143

Figure 23. Sample output of the service application developed for the gene linkage application

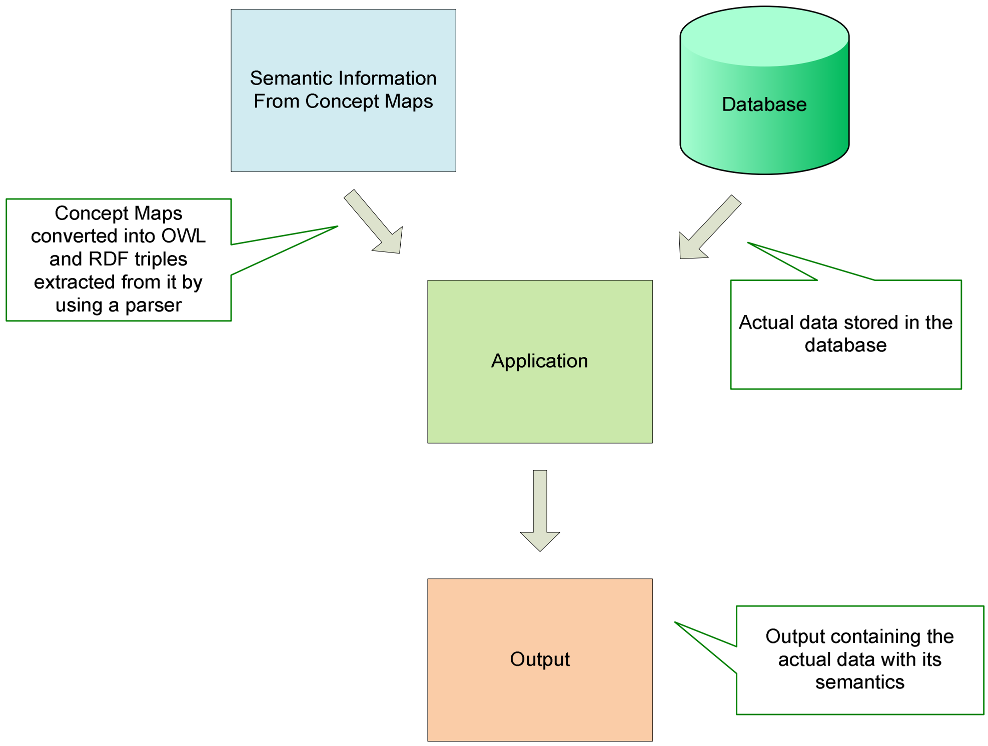


Figure 24. Architecture of the gene linkage application

The development of tools like the gene linkage application can be useful in enhancing the speed of research activities in many areas. Because the PAAC Framework provides access to the semantic information associated with the data, the learning curve of the researcher is reduced. Some of the key contributions of this application are as follows:

i) Provides a proof-of-concept for building applications (uses the idea of representing a process-as-a-concept).

ii) Provides the user with data combined with their semantics.

iii) Allows the domain expert to represent semantics of the terminologies involved in gene linkage.

iv) Reduces the need for the user to have prior knowledge of terminologies involved in gene linkage.

D.1 Validating the Concept Map

The concept map depicted in Figure 19 is validated by using Table 12 and the table contained in Figure 22. Table 12 depicts the RDF triples built by the developer while working with the domain expert. The table contained in Figure 22 is generated by the Java parser. Confirming the validity of the concept map depicted in Figure 19 requires that these two tables match one another.

E. Automation of Concept Map Creation in Medical Research

Even with current advances in technology, the processes involved in medical research continue to be both time consuming and labor intensive. To mitigate these problems we have built an integrated tool for converting the textual information into a concept map available to the researchers; this conversion uses OWL as an intermediate source of information. This tool is based on building semantic models by using concept maps. The labor intensive sequence of processes involved in medical research is effectively replaced by using this tool built by a suitable integration of concept maps and OWL. The tool which has been analyzed by considering the example of linking Vitamin D deficiency to prostate cancer [44], is intended to provide a faster solution in building relations and concepts based on the existing facts.

E.1 Challenges to Finding the Right Research Material in Medical Research

A typical researcher spends an inordinate amount of time in identifying, sorting, and analyzing already published research results [62]. This time can be lessened by several weeks with the use of appropriate tools and technologies [63]. The problem of inefficiency in medical research results from the following reasons:

i) Inefficiencies occur when researchers use multiple databases with multiple interfaces that may not all be known by a research assistant.

ii) Even when a research assistant is well versed on the search techniques, he or she may not be confident that all relevant research papers have been identified.

iii) When relevant research papers are identified, the next challenge is to efficiently extract information on materials, methods, and results illustrated in the papers.

These problems cause a significant delay between a researcher's obtainment of results and use of those results. A significant increase also occurs between the time at which most researchers form a consensus and the time at which such findings reach to practicing physicians. Even if a given research result is available, practicing physicians find it difficult to stay abreast of such research because of their time limitations [64]. Even if a physician reserves a certain amount of each day to keep up with current research, he or she will face hundreds of articles on a given topic and will be able to read only a handful of them. Months if not years are required for research findings to reach to practitioners. A case in point is the research on Vitamin D [65]; because this research has been ongoing for more than 30 years and has increased in the last few years, many research findings on Vitamin D are available for integration into the practices of physicians who do not have sufficient time to read the reports containing these findings.

Medical researchers commonly use an evidence-based methodology [66] for building the results of a particular medical research. This methodology involves the three-step approach as indicated in Figure 25 and is described as follows:

Step 1: Collection of data, knowledge, and experience from clinical studies.

Step 2: Analysis of the data collected in Step 1.

Step 3: Presentation of the evidence based on the analysis of data.

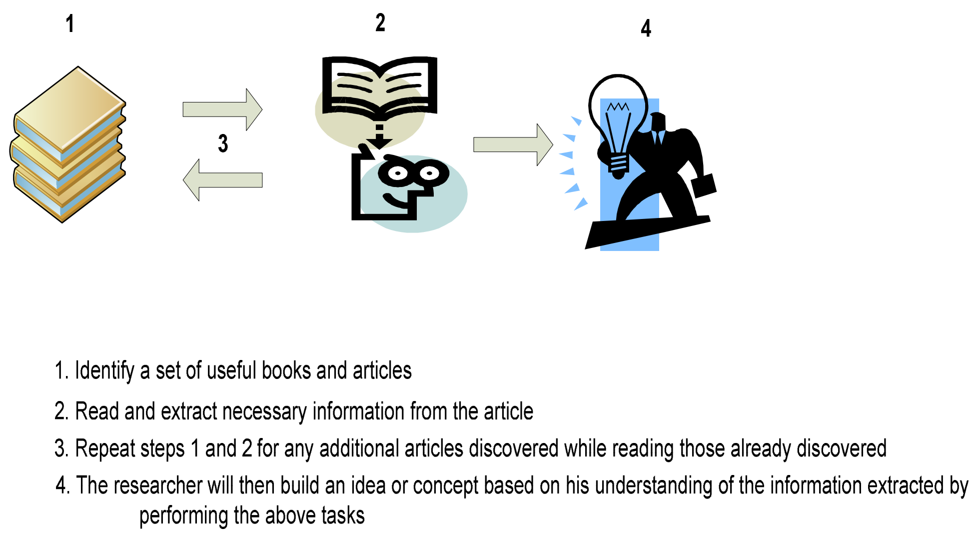


Figure 25. Conventional medical research

Sometimes, two or more distinct groups of researchers, investigating the same subject may conclude with contradictory results despite following the steps [67] [68] [69]. The United States Preventive Services Task Force (USPSTF) provides guidance for the clinicians by grading its recommendations according to one of five classifications (A, B, C, D, and I) reflecting the strength of evidence and magnitude of net benefit [70]. The current state of the art in accessing available published research findings, along with the complexity of the levels of evidence required by researchers and clinicians, demonstrates the need for improvement in medical research techniques [71].

To solve the problem of finding and searching through published research work, one must address the underlying technological issues [72]. Multi-platforms with multi-interface requirements can be addressed with the concept of tools as services. Under such an approach, the researcher will use the tool that he or she always uses, such as an Excel spreadsheet. The software behind the tool automatically searches for the data requested by the researcher and populates the Excel spreadsheet with the required data. In this manner, a researcher does not waste time in learning unnecessary interfaces but continues to use the tool with which he or she is familiar.

The problem of determining whether all published information is identified can be alleviated by semantic search tools that perform "deep searches" that may not be accessible to commonly used tools such as Google. The problem of preparing comprehensive information that can be used to update research assistants or practicing physicians can be solved by concept maps discussed in the next section.

E.2 Integrating Concept Maps and Ontology

Researchers can use C-Map Tools software to represent an idea or a concept in a graphical format that links all concepts and resembles a growing neural network. This tool allows the conceptual information thus obtained to be automatically converted into OWL and RDF, as needed, in preparation for further processing. In fact, this robust, domain-specific information stored in OWL can be readily manipulated by the rule-based engine of the Jena Semantic Web Framework, as well as accessed by intelligent agents using Semantic Web technology for automated inference capability.

A critical challenge for storing domain knowledge of a particular medical research area involves storing this information in a human perceivable as well as machine-actable format. Concept maps can fulfill both of these requirements. Concept maps can delineate a concept in the form of links and circles, as well as allow these concepts to be converted into XML-based ontology representation such as OWL and vice versa [44]. Achieving the accumulation of domain knowledge in the form of concepts requires completing (Figure 26):

i) Identify the electronic documents that pertain to the topic of medical research and its effects.

ii) Convert the information contained in the document into an XML-based format.

iii) Convert the information available in the XML-based format into a concept.

The concepts thus obtained can be further converted into OWL or other forms of Semantic Web-based format. This conversion would allow the extraction of relevant information by an application program that can harness the versatility of Semantic Web-based information [73]. Overall, this process provides a path to conversion of natural language-based domain information into machine-actable information that can be readily used by an application program. This process enables the achievement of the following objectives:

i) It saves the amount of time required to decipher information available on a particular topic of medical research [63].

ii) It automates the gathering of organized information and thereby provides a suitable path for linking its deficiency to diseases.

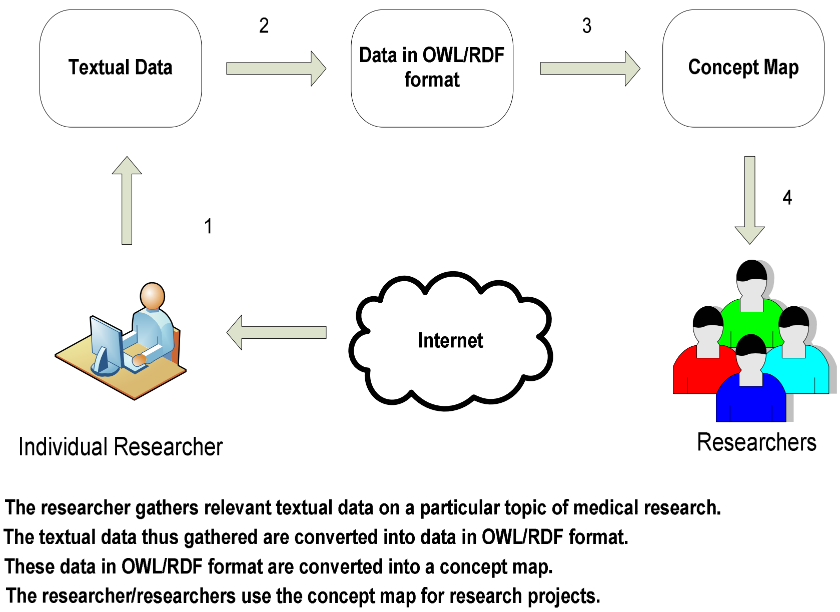


Figure 26. The process of converting textual data into a concept map

Note: From "Building semantic models using concept maps for medical research," by V. P. Gurupur, M. N. Tanju, T. Tuncer, C. C. Gattaz, and M. M. Tanik, *Proc. 12th SDPS Transdisciplinary Conf. –Workshop on Integrated Systems, Design and Process Science*, Montgomery, AL, 2009. Copyright 2009 by Society for Design Process Science (SDPS). Adapted with permission.

E.3 Relating Vitamin D Deficiency to Prostate Cancer

To provide an example of a proof of concept, we consider the analysis of the relationship between Vitamin D deficiency and prostate cancer [74] [75]. Vitamin D is an important calcium homeostasis regulator that also modulates growth and differentiation. There seems to be a highly complex relationship between serum levels of Vitamin D and the incidence of prostate cancer. Some studies [76] [77] have linked calcium intake to an elevated risk of prostate cancer; in those studies, it was assumed that this effect was primarily through lowered serum levels of α1, 25-dihydroxyvitamin D (calcitriol) and that these lowered levels resulted from decreases in parathyroid hormone levels. Assuming that Vitamin D maintains the differentiated state of prostate cells, researchers made multiple attempts to determine the therapeutic potential of Vitamin D and/or its analogs in men with castrate-resistant prostate cancer.

Vitamin D receptors have been identified on most but not all prostate cancer cell lines. Exposure to calcitriol induces differentiation, as well as decreased proliferation, invasiveness, and metastatic behavior [78][79][80]. Animal studies provide support for an antiproliferative and antimetastatic role of calcitriol and its analogs in prostate cancer [81]. Calcitriol potentiates the cytotoxic effect of some chemotherapeutic agents like taxanes, and this finding provides a rationale for combined therapy [82] for metastatic disease.

In terms of the role of Vitamin D in preventing prostate cancer [83], epidemiologic studies suggest that the relationship between Vitamin D levels and the incidence of prostate cancer is extremely complex. Vitamin D deficiency has been suggested as a common pathway underlying the association of prostate cancer risk with other epidemiologic risk factors such as, age, African American race, and geographic area of residence [84]. Ahn and co-authors [85] state that "no statistically significant trend in overall prostate cancer risk was observed with increasing season-standardized serum 25(OH)D level." Because the data relating Vitamin D status to prostate cancer is sparse and inconsistent and because achieving a meaningful conclusion for providing recommendations requires a complex approach, building an unconventional method of research is inevitable

[44]. To achieve this, we propose the use of concept maps to delineate the complex research observations.

E.4 Using the Concept Map Alternative

A plausible solution to the problem of relating Vitamin D to prostate cancer is to build a knowledge base of Vitamin D and its effect on prostate cancer. Building such a base can be achieved by using a combination of tools involving concept maps. If this approach is used, the tasks involved in searching research findings could be obviated by building a set of concept maps from selected articles. These maps would also considerably reduce the amount of time and effort used by a researcher to carry out research activities. Concepts built in the human mind can now be displayed in the form of a visual description available to scientists. To achieve this, we have performed the following set of tasks:

i) Identify the electronic documents that pertain to Vitamin D and indicate that its deficiency leads to prostate cancer.

ii) Convert the information contained in the documents into an OWL document by using a tool such as TopBraid Composer [86], developed by Franz Inc. [87].

iii) Use CMap Tools COE [88] to convert the OWL document into a concept map.

The concept map obtained by performing these three tasks can be easily used to depict the relationship among terminologies, concepts, and procedures related to Vitamin D. Sometimes the investigators can populate the research data on an Excel spreadsheet, and converting this application into a semantic model might prove useful. The three steps can also be used to convert the data available on a spreadsheet.

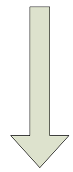
Overall, we have presented an integrated tool with which to convert textual data into concept maps. While the CMap Tools software from IHMC has the ability to convert any document in OWL into a concept map, using TopBraid Composer conversion of textual data into OWL is performed. It is to be noted that, although this process provides a path for building concept maps from textual data, building an understandable concept map requires refactoring the data in OWL. The key contributions of this application are as follows:

i) Demonstrates a means by which textual data can be converted into concept maps by integrating the required tools.

ii) Demonstrates a path for applying concept maps to improve an area of medical research by reducing the amount of time and labor involved in carrying out complicated research.

Figures 27 and 28 provide a path by which a simple text such as “Vitamin D deficiency leads to prostate cancer” can be depicted in the form of a diagram. Concept maps generated by IHMC CMap Tools software are used to generate this path.

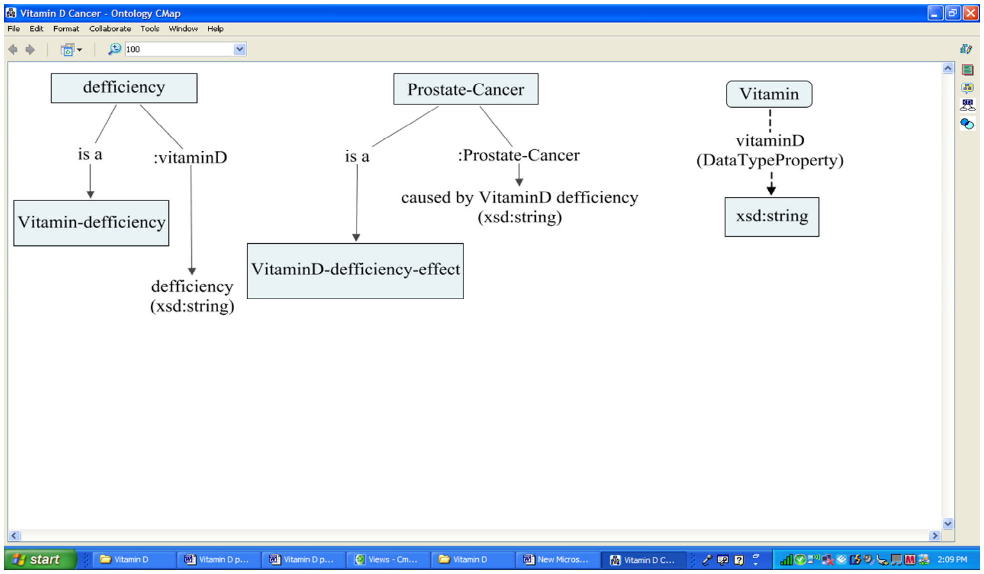
Vitamin D deficiency leads to Prostate Cancer




```xml
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://www.mycompany.com/TopBraid/Vit_Pro_Cancer.owl#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://www.mycompany.com/TopBraid/Vit_Pro_Cancer.owl">
  <owl:Ontology rdf:about="">
    <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Created with TopBraid Spreadsheet converter</owl:versionInfo>
  </owl:Ontology>
  <owl:Class rdf:ID="Vitamin"/>
  <owl:DatatypeProperty rdf:ID="vitaminD">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Vitamin D </rdfs:label>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#Vitamin"/>
  </owl:DatatypeProperty>
  <Vitamin-defficiency rdf:ID="defficiency">
    <vitaminD rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >defficiency</vitaminD>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >defficiency</rdfs:label>
  </Vitamin-defficiency>
  <VitaminD-defficiency-effect rdf:ID="Prostate-Cancer">
    <Prostate-Cancer rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >caused by VitaminD defficiency</Prostate-Cancer>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >defficiency cause</rdfs:label>
  </VitaminD-defficiency-effect>
</rdf:RDF>
```


Figure 27. Converting a text into Web Ontology Language

Note: From “Building semantic models using concept maps for medical research,” by V. P. Gurupur, M. N. Tanju, T. Tuncer, C. C. Gattaz, and M. M. Tanik, *Proc. 12th SDPS Transdisciplinary Conf. –Workshop on Integrated Systems, Design and Process Science*, Montgomery, AL, 2009. Copyright 2009 by Society for Design Process Science (SDPS). Adapted with permission.




```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://www.mycompany.com/TopBraid/Vit_Pro_Cancer.owl#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://www.mycompany.com/TopBraid/Vit_Pro_Cancer.owl">
  <owl:Ontology rdf:about="">
    <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Created with TopBraid Spreadsheet converter</owl:versionInfo>
  </owl:Ontology>
  <owl:Class rdf:ID="Vitamin"/>
  <owl:DatatypeProperty rdf:ID="vitaminD">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Vitamin D </rdfs:label>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#Vitamin"/>
  </owl:DatatypeProperty>
  <Vitamin-defficiency rdf:ID="defficiency">
    <vitaminD rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >defficiency</vitaminD>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >defficiency</rdfs:label>
  </Vitamin-defficiency>
  <VitaminD-defficiency-effect rdf:ID="Prostate-Cancer">
    <Prostate-Cancer rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >caused by VitaminD defficiency</Prostate-Cancer>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >defficiency cause</rdfs:label>
  </VitaminD-defficiency-effect>
</rdf:RDF>
```

Figure 28. Mapping the information from a Web Ontology Language document into a concept map

Note: From "Building semantic models using concept maps for medical research," by V. P. Gurupur, M. N. Tanju, T. Tuncer, C. C. Gattaz, and M. M. Tanik, *Proc. 12th SDPS Transdisciplinary Conf. –Workshop on Integrated Systems, Design and Process Science*, Montgomery, AL, 2009. Copyright 2009 by Society for Design Process Science (SDPS). Adapted with permission.

F. Conclusion

In this chapter, the dissertation presents a solution for the problem stated in Chapter 1. It provides a solution by designing a framework for representing a process as a concept. This framework, termed as PAAC Framework, has been validated by building the gene linkage application. Additionally, this chapter discusses some experimentation on the automation of concept map creation; this experimentation has further advanced the art of using concept maps to represent a process.

CHAPTER 4

CONCLUSION

A. Key Contributions of PAAC Framework

The PAAC Framework provides a unique method of converting processes into concepts. The core contributions of this dissertation are as follows:

i) Reduction in the semantic gap involved in the process of Semantic Service Composition.

ii) A framework for presenting process-as-a-concept in the context of semantic service composition.

iii) Automation of building concept maps from textual data.

iv) A proof-of-concept that this framework is sound, valid, and applicable is shown by providing the gene linkage application.

A.1 Reduction in the Semantic Gap

In this dissertation, the proposed PAAC Framework was shown to reduce the semantic gap involved in the process of developing composite services. The representation of a process by using abstract process models is not an adequate solution because the abstract models may not clearly present the user processes. Some computer users may not be able to easily comprehend a conventional abstract process model used to represent the processes involved in an enterprise system. The representation of a user process by using a task system model provides both the user and the developer the ability to construe the representation of user processes. As a result, the user can provide the developer with feedback about the understanding and representation of the user processes.

The transformation of the task system model into concept maps facilitates the development of composite services because the concept maps thus obtained can be converted into machine-actable ontologies. These machine-actable ontologies derived from concept maps can be easily parsed by using a suitable parser and therefore can enhance the process of selecting appropriate services.

A.2 Development of the PAAC Framework

In this dissertation, an analysis of evaluating various methodologies and technologies for potential integration into the software service composition used to fill the semantic gap between software requirements and implementation is presented. The usefulness of this approach has been described with a case study demonstrating the development of a gene linkage application. This framework will be further enhanced by introducing a host of advanced methodologies and technologies useful for integrating software services and especially by using ontologies to exploit the model-driven approach to software design using ontologies. Overall, the PAAC Framework, which saves time and effort during the process of software development, presents a path for the future development of monolithic and complex software systems [89] [90].

A.3 Automation of Building Concept Maps from Textual Data

In this dissertation, we provided a presentation using a concept map to depict a particular concept in medical research. The dissertation also demonstrates the means by which textual data can be converted into concept maps by integrating the required tools to achieve this task. In this way, the dissertation has provided a path for applying new visual tools with which to improve the area of medical research by reducing the amount of time and labor involved in carrying out complicated tasks.

To illustrate a unique method for accelerating the process of medical research, we included in this dissertation a literature search on the subject of carcinogenesis prevention. There is a great need to improve current ways of research methods by using technologies such as concept maps and the Semantic Web. The proposed approach involves the usage of rapidly evolving cutting edge technologies such as CMap Tools COE that can revolutionize medical research. The dissertation has provided a path by which future generations of medical researchers can improve the techniques used to carry out research activities pertaining to a disease or disorder.

A.4 Proof-of-Concept for the PAAC Framework

The gene linkage application depicted in Chapter 3 provided the required proof-of-concept for the PAAC Framework. This dissertation clearly indicates the means by which this application is built within the perimeters of the framework.

B. Comparison with Other Methods and Frameworks

B.1 Comparison with CommonKADS

In Chapter 2, the dissertation describes CommonKADS methodology and its use in representing information about processes involved in an enterprise system. Although an elegant system for representing knowledge of an organization, CommonKADS does not provide any insight into its use in building software applications. Moreover, CommonKADS provides a complex representation of the processes involved in an organization. Alternatively, PAAC Framework uses a simple representation of a process and provides the necessary information on building a software system. Table 14 provides a comparison between the PAAC Framework and CommonKADS.

Table 14.

Comparison with CommonKADS

Process-as-a-Concept Framework	CommonKADS
Provides implementation details of using the information of a process stored in the form of concept maps	Emphasizes only on the representation and storage of knowledge
Use of a simple task system model to represent a process	Use of complex models to represent tasks and processes

B.2 Comparison with Bio2RDF

As discussed in Chapter 2, Bio2RDF is a knowledge system for bioinformatics built by using the semantic web technology. Some similarities exist between the PAAC Framework and Bio2RDF because both of them rely on the usage of semantic web technology. However, although Bio2RDF provides implementation details on storing and retrieving information relevant to bioinformatics useful for building software systems, the PAAC Framework provides both abstract and machine-actable representation of information to achieve the same.

Incidentally, the PAAC Framework is software centric, whereas Bio2RDF can be both software centric and user centric, because the Bio2RDF system allows a user to query its knowledge base by using a SPARQL query interface [50]. Some of the main differences between the PAAC Framework and Bio2RDF are given in Table 15.

Table 15.

Comparison with Bio2RDF

Process-as-a-Concept Framework	Bio2RDF
Uses concept maps to represent domain information	Uses text or XML documents to represent domain information
Facilitates easily understandable representation of abstract domain information	Does not facilitate conceptual representation of abstract information
Software centric: Only used for building software systems	Both user centric and software centric

B.3 Comparison with CP^2 Framework

The PAAC Framework is built on the underlying principles of the CP^2 Framework developed by Sadasivam [15]. The critical difference between the CP^2 Framework and the PAAC Framework is the involvement of semantic information associated with the data and the services. The CP^2 Framework follows the conventional approach of using the abstract process model and transforming it into an XML-based executable process model. The abstract process model used in the conventional approach is complex and difficult to comprehend, whereas the use of concept maps reduces the complexity involved in representing a process. Additionally, the machine-actability achieved by using XML-based languages such as OWL can be easily parsed by using a semantic web-based toolkit. Table 16 lists the differences between the PAAC Framework and the CP^2 Framework.

Table 16.

Comparison with CP^2 Framework

Process-as-a- Concept Framework	CP^2 Framework
This framework involves two level representation of domain knowledge information by using concept maps	This framework develops processes for composition in process-oriented composite services
Level 1: Concept maps allow the domain expert to build a schematic representation of the domain knowledge	Abstract Process Models – Starts with high level models with few details. As details are added, abstract process models can be transformed to executable process models
Level 2: The concepts can be converted into machine-actable XML-based format that can be used by programs capable of parsing and utilizing this information	Executable Process Models are models that can be directly composed on a composition engine, e.g., BPEL

Note: From "Representing processes as concepts: Towards reducing semantic gap," by V. P. Gurupur and R.S. Sadasivam, *Proc. 12th SDPS Transdisciplinary Conf. –Workshop on Integrated Systems, Design and Process Science*, Montgomery, AL, 2009. Copyright 2009 by Society for Design Process Science. Adapted with permission.

C. Limitations of the PAAC Framework

Although providing several advantages in filling the semantic gap in building composite services, the PAAC framework possesses limitations. These limitations are as follows:

i) The usage of the task system model limits it to resource-centric processes.

ii) The task system model may not be adequate to represent complex processes.

iii) The aspect of communication between processes is not represented adequately in PAAC.

iv) The concept maps may not represent sufficient information on some processes.

C.1 Resource-Centric Processes

The dissertation illustrates the functioning of the PAAC Framework by using resource-centered processes. Its functionality in processes that do not heavily depend on a resource or a set of resources has not yet been analyzed. The task system model is inherently resource centric and this characteristic creates a limitation in the application of the framework.

C.2 Complex Processes

The application of the framework to complex processes has not yet been analyzed. An observation has been made that methods such as CommonKADS can adequately be used to represent a wide range of processes. However, the task system model may not adequately represent processes that are complex and can be represented by using CommonKADS.

D. Future Work

D.1 Building Systems-as-a-Concept Approach

The induction of a systems approach towards building a concept may be considered for future work. Such an approach would replace the representation of a process with the representation of a system. Therefore, the representation of a process by using the task system model would be replaced by the representation of a system by using a language such as SysML. This systems approach will bring about a wider applicability of the philosophy of using a conceptual representation. The representation of a software system by using concept maps results in an abstract and machine-actable representation of the system. A critical challenge in building a system-as-a-concept approach would be to develop a methodology for converting a SysML representation into concept maps. Because SysML facilitates the representation of a broad range of systems, the conversion of a SysML representation to a concept map can be challenging.

D.2 Building Commercial Applications by Using the PAAC Framework

The PAAC Framework can be used for building various commercial applications. The primary objectives of this dissertation were to introduce this framework and to provide a proof-of-concept that it for its use in building composite services. Because the framework is still in its embryonic stage, some improvements may be required before it can be used to build a wide range of commercial applications.

E. Summary

This dissertation provides a framework for building composite services by reducing the problem of the semantic gap. The key features of this dissertation are as follows:

i) The need for reducing semantic gap in the process of Semantic Service Composition is established.

ii) The framework of representing processes as concepts is introduced as a means of reducing this semantic gap.

iii) A proof-of-concept application for gene linkage is provided to validate this framework.

iv) An additional application for automating the generation of concept map is provided.

v) The semantic gap is shown to be greatly reduced by representing processes as concepts when composing semantic services.

As indicated in Section D, this dissertation provides intension of the author to continue this research on building concept maps for software systems. The author perceives this dissertation work as a starting point on the road toward building semantic web-based software development approaches.

LIST OF REFERENCES

[1] P. Ganguly, F.A. Rabhi, and P.K. Ray, “Bridging semantic gap,” in *Proc. 2002 Conf. Pattern Languages Programs,* Melbourne, 2003, pp. 59–61.

[2] V. P. Gurupur and R. S. Sadasivam, “Representing processes as concepts: Towards reducing semantic gap,” in *Proc. 12th SDPS Transdisciplinary Conf. –Workshop on Integrated Systems, Design and Process Science*, Montgomery, AL, 2009 *(CD ROM).*

[3] N. Chomsky. (1956). Three models for the description of a language [Online]. Available: http://www.chomsky.info/articles/195609--.pdf. Accessed: 02/02/2010.

[4] R. L. Glass, *Software Runaways*. NJ: Prentice-Hall, 1998.

[5] V. Gurupur and M. M. Tanik, “A System for Building Clinical Research Applications using Semantic Web-Based Approach,” *Journal of Medical Systems,* DOI: 10.1007/s10916-010-9445-8.

[6] J. Greenfield and K. Short, *Software Factories*. Indianapolis, IN: Wiley, 2004, pp. 79 –81.

[7] S. R. Schach, *Object Oriented and Classical Software Engineering*, 6th ed. New York: McGraw-Hill, 2005, pp. 48–57.

[8] S. A. Bernard, *An Introduction to Enterprise Architecture*, 2nd ed. Bloomington, IN: AuthorHouse, 2005, pp. 25–31.

[9] D. Allemang and J. Hendler, *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. Burlington, MA: Elsevier, 2008, pp. 31–51.

[10] (2009). Preliminary SysML RFI analysis presented to SE DSIG in Long Beach. OMG Systems Modeling Language. Available: http://www.omgsysml.org. Accessed: 02/09/2010.

[11] M.M. Tanik and E.S. Chan, *Fundamentals of Computing for Software Engineers*. Van Nostrand Reinhold, 1991.

[12] Axiomatic Design. (2005). Slideshow on Axiomatic Design and Acclaro software. [Online]. Available: http://www.vr.clemson.edu/credo/classes/lect6-15.pdf#search='axiomatic%20design. Accessed: 09/09/2009.

[13] R. M. Gray. (2009, July,16). Entropy and Information Theory. [Online]. Available: http://ee.stanford.edu/~gray/it.pdf. Accessed: 09/21/2009.

[14] (2009, January). Latent semantic analysis. [Online]. Available: http://lsa.colorado.edu/. Accessed: 09/21/2009.

[15] J. Cardoso. *Semantic Web Services: Theory, Tools and Applications*. Hershey, PA: IGI Global, 2007, pp. 44–71.

[16] W3C Definition of a Web Service. (2004). Web services [Online]. Available: http://www.w3.org/TR/ws-gloss/#ssem. Accessed: 12/20/2009.

[17] S. Dustar. (2003). Service composition [Online]. Available: http://www.unitn.it/convegni/download/icsoc03/panel/P_P2_Dustdar.pdf. Accessed: 12/21/2009.

[18] R. S. Sadasivam, "An architecture framework for composite services with process personalization," Ph.D. dissertation, Dept. Elect. Comput. Eng.; University of Alabama at Birmingham, Birmingham, AL, 2007.

[19] G. Kotonya and I. Sommerville, *Requirements Engineering: Processes and Techniques*. Indianapolis, IN: Wiley, 2001, pp. 113–136.

[20] K. E. Wiegers, *Software Requirements*, 2nd ed. Microsoft Press, 2003, pp. 3–27, 131–153.

[21] International Council on Systems Engineering (INCOSE) [Online]. Available: http://www.incose.org/. Accessed: 01/23/2010.

[22] A. Newell. (1980, August 19). The knowledge level [Online]. Available: http://lacam.di.uniba.it:8000/people/courses/kr/AIMag02-02-001.pdf. Accessed: 03/11/2010.

[23] C. A. Welty., "An integrated representation for software development and discovery," Ph.D. dissertation, Dept. Comput. Sci.; Rensselaer Polytechnic Institute, Troy, NY, 1995.

[24] I. Sommerville, *Software Engineering*, 8th ed. Harlow, Essex: Addison-Wesley, 2006.

[25] G. Schreiber, H. Akkermans, A. Anjewierden, R. Hoog, N. Shadbolt, W. V. Velde and B. Wielinga, *Knowledge Engineering and Management: The CommonKADS Methodology*. Cambridge, MA: The MIT Press, 2000.

[26] CommonKADS Website. [Online]. Available: http://www.commonkads.uva.nl/frameset-commonkads.html. Accessed: 02/02/2010.

[27] U. J. Tanik, “Artificial intelligence design framework,” Ph.D. dissertation, Dept. Elect. Comput. Eng., University of Alabama at Birmingham, Birmingham, AL, 2007.

[28] J. D. Novak and A. J. Cañas (2006). The theory underlying Concept Maps and how to construct and use them [Online]. Available: http://cmap.ihmc.us/Publications/ResearchPapers/TheoryUnderlyingConceptMaps.pdf. Accessed: 12/26/2009.

[29] A. J. Cañas, J. D. Novak, and F. M. González (2004). Two-layered approach to knowledge representation using conceptual maps and description logics [Online]. Available: http://cmc.ihmc.us/papers/cmc2004-205.pdf. Accessed: 12/26/2009.

[30] T. Buzan, *Mind Maps at Work*. New York: Plume, 2005.

[31] History of mind mapping [Online]. Available: http://www.mindmeister.com/content/historyofmm. Accessed: 02/09/2010.

[32] Object Management Group Website [Online]. Available: http://www.omg.org/gettingstarted/gettingstartedindex.htm. Accessed: 02/02/2010.

[33] SysML FAQ [Online]. Available: http://www.sysmlforum.com/FAQ.htm. Accessed: 01/22/2010.

[34] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide*, 2nd ed. Reading, MA: Addison-Wesley, 1998, pp. 13–58.

[35] SysML Website [Online]. Available: http://www.omgsysml.org. Accessed: 01/21/2010.

[36] J. Greenfield, K. Short, S. Cook, and S. Kent, *Software Factories*. Indianapolis, IN: Wiley, 2004.

[37] S. F. Mills and M. M. Tanik, “Resource-focused process engineering for the distributed enterprise,” *Int. Comput. Integrated Mfg*, vol. 13, No. 3, pp. 187–203, May 2000.

[38] E. G. Coffman and P. J. Denning, *Operating Systems Theory*. Prentice-Hall, 1973, pp. 35.

[39] E. Newcomer and G. Lomow, *Understanding SOA with Web Services*. Upper Saddle River, NJ: Addison-Wesley, 2005.

[40] W3C Semantic Web Definition [Online]. Available: http://www.w3.org/2001/sw/. Accessed: 01/21/2010.

[41] M. C. Daconta, L. J. Obrst, and K. T. Smith, *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management.* New York: Wiley, 2003, pp. 56–71.

[42] RDFPic Application. [Online]. Available: http://jigsaw.w3.org/rdfpic/rdfpic-2/rdfpic-readme.html. Accessed: 12/16/2009.

[43] OWL Website [Online]. Available: http://www.w3.org/TR/owl-features. Accessed: 12/01/2009.

[44] V. P. Gurupur, M. N. Tanju, T. Tuncer, C. C. Gattaz, and M. M. Tanik, "Building semantic models using concept maps for medical research," in *Proc. 12th SDPS Transdisciplinary Conference–Workshop on Integrated Systems, Design and Process Science*, Montgomery, AL, 2009, *(CD ROM).*

[45] IBM, BPEL4WS Specifications [Online]. Available: http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel/ws-bpel1.pdf. Accessed: 01/21/2010.

[46] UMLS Website [Online]. Available: http://www.nlm.nih.gov/ research/umls/about_umls.html. Accessed: 01/25/2010.

[47] DBpedia Website [Online]. Available: http://dbpedia.org/. Accessed: 02/02/2010.

[48] J A. Zachmann, Framework Definition [Online]. Available: http://esvc000904.wic047u.server-web.com/PDF-papers/zachman1.pdf. Accessed: 01/21/2010.

[49] Federal Information Architecture Website [Online]. Available: http://www.gao.gov/bestpractices/bpeaguide.pdf. Accessed: 01/22/2010.

[50] M. Jarrar and R. Meersman, "Formal ontology engineering in the DOGMA approach," in *Proc. Intl. Conf. Ontologies, Databases and Applications of Semantics,* CA, 2002, pp. 1238–1254.

[51] DoDAF Website [Online]. Available: http://cio-nii.defense.gov/ docs/DoDAF_Volume_II.pdf. Accessed: 02/01/2010.

[52] Bio2RDF Website [Online]. Available: http://bio2rdf.org. Accessed: 02/03/2010.

[53] F. Belleau, M. A. Nolin, N. Tourigny, P. Rigault, and J. Morissette. "Bio2RDF: towards a mashup to build bioinformatics knowledge systems," *J. of Biomed. Informatics* [Online]. *41(5)*. pp. 706–716. Available: http://mashup.pubs.dbs.uni-lzig.de/files/ Belleau2008BioRDFTowardsamashuptobuildbioinformaticsknowledge.pdf. Accessed: 03/10/2010.

[54] MIT, SIMILE Project [Online]. Available: http://simile.mit.edu/RDFizers. Accessed: 02/02/2010.

[55] UAB Department of Epidemiology Website [Online]. Available: https://www.soph.uab.edu/index.php. Accessed: 02/06/2010.

[56] NCBI Website [Online]. Available: http://www.ncbi.nlm.nih.gov/. Accessed: 12/27/2009.

[57] PubMed Website [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed. Accessed: 02/08/2010.

[58] OMIM NCBI Website [Online]. Available: http://www.ncbi.nlm.mih.gov/omim. Accessed: 01/10/2010.

[59] A. Chowdhury and P. Choudhary, *JAX: Java APIs for XML Kick Start*. Sams, 2003.

[60] D. Chappell, *Understanding .NET*, 2nd ed. Boston, MA: Pearson Education, 2002.

[61] H. Bergsten, *Java Server Pages*, 3rd ed. Sebastopol, CA: O' Reilly, 2004.

[62] T. Calishain, *Information Trapping: Real-Time Research on the Web*. Berkeley, CA: New Riders, 2007, pp. 1–6.

[63] B. Hu. (2008, March 11). Semantic Web technologies can help save lives [Online]. Available: http://www.semantic-web.at/1.36.resource.265.bo-hu-x22-semantic-web-technologies-can-help-save-lives-x22.htm. Accessed: 08/11/2009.

[64] K. Yamamoto, M. Matsumoto, H. Tada, K. Yanagihara, S. Teramukai, T. Takemura, and M. Fukushima, "A data capture system for outcomes studies that integrates with electronic health records: Development and Potential Uses", *Journal of Medical Systems,* vol. 32, pp. 423–427, May 2008.

[65] F. Murray, *Sunshine and Vitamin D: A Comprehensive Guide to the Benefits of the "Sunshine Vitamin."* Laguna Beach, CA: Basic Health Publications, 2008.

[66] R. C. Brownson, E. A. Baker, T. L. Leet and K. N. Gillespie, *Evidence-Based Public Health*. New York: Oxford University Press, 2003, pp. 5–6.

[67] J. P. Ioannidis. (2005, July 13). Contradicted and initially stronger effects in highly cited clinical research. *MEDLINE* [Online]. *294(2).* pp. 218–228. Available: http://www.mdconsult.com/das/citation/body/1519448152/jorg=journal&source=MI&sp=15634591&sid=0/N/15634591/1.html?issn=. Accessed: 08/27/2009.

[68] NHMRC additional levels of evidence and grades for recommendations for developers of guidelines [Online]. Available: http://www.nhmrc.gov.au/guidelines/consult/consultations/add_levels_grades_dev_guidelines2.htm. Accessed: 08/11/2009.

[69] T. Merlin, A. Weston, and R. Tooher. (2009, June 11). Extending an evidence hierarchy to include topics other than treatment: Revising the Australian 'levels of evidence'. *BMC Medical Research Methodology* [Online]. Available: http://www.biomedcentral.com/1471-2288/9/34#B1. Accessed: 08/11/2009.

[70] U.S. Preventive Services Task Force Ratings: Strength of Recommendations and Quality of Evidence. (2000–2003). *Guide to Clinical Preventive Services*, (3rd ed.) [Online]. Available: http://www.ahrq.gov/clinic/3rduspstf/ratings.htm. Accessed: 08/11/2009.

[71] A. Eren, A. Subasi and O. Coskun, "A decision support system for telemedicine through the mobile telecommunications platform," *J. Med. Systems,* vol. 32, pp. 31–35, Feb. 2008.

[72] S-H. Hsieh, I-C. Hou, P-H. C-T. Cheng, Tan, P-C. Shen, K-P. Hsu, S-L. Hsieh and F. Lai. (2009, May 22). Design and implementation of a web-based mobile elec tronic medication administration record. *J. Med. Systems* [Online]. Available: http://www.springerlink.com/content/e537234j8h7q0430/fulltext.pdf. Accessed: 08/07/2009.

[73] D. Collett, *Modeling Survival Data in Medical Research*, 2nd ed. Boca Raton, FL: CRC Press, 2009.

[74] A. Zittermann, "Vitamin D in preventive medicine: Are we ignoring the evidence?" *Bri. J. Nutr.,* vol. 89, pp. 552–572, May 2003.

[75] H.W. Lim, H. Hönigsmann and J. Hawk, *Photodermatology*. New York: Informa Healthcare, 2007, pp. 41–118.

[76] E. Giovannucci, "Dietary influences of 1,25(OH)2 vitamin D in relation to prostate cancer: A hypothesis," *Cancer Causes and Control*, vol. 9, pp. 567–582, Dec. 1998.

[77] J. M. Chan, E. Giovannucci, S. O. Andersson, J. Yuen, H. O. Adami, and A.Wolk,"Dairy products, calcium, phosphorous, vitamin D, and risk of prostate cancer (Sweden)," *Cancer Causes and Control*, vol. 9, pp. 559–566, Dec. 1998.

[78] A.V. Krishnan, D. M. Peehl, and D. Feldman, "Inhibition of prostate cancer growth by vitamin D: Regulation of target gene expression," *J. Cellular Biochem.*, vol. 88, pp. 363–371, Feb. 2003.

[79] M. Guzey, S. Kitada, and J. C. Reed, "Apoptosis induction by 1alpha,25-dihydroxyvitamin D3 in prostate cancer," *Molecular Cancer Therapeutics,* vol. 1, pp. 667–677, Jul. 2002.

[80] S. H. Zhuang and K. L. Burnstein, "Antiproliferative effect of 1alpha,25-dihydroxyvitamin D3 in human prostate cancer cell line LNCaP involves reduction of cyclin-dependent kinase 2 activity and persistent G1 accumulation," *Endocrinology*, vol. 139, pp. 1197–1207, Mar. 1998.

[81] W. Banach-Petrosky, X. Ouyang, H. Gao, K. Nader, Y. Ji, N. Suh, R. S. DiPaola, and C. Abate-Shen, "Vitamin D inhibits the formation of prostatic intraepithelial neoplasia in Nkx3.1;Pten mutant mice," *Clin. Cancer Res.*, vol. 12, pp. 5895–5901, Oct. 2006.

[82] T. M. Beer, K. M. Eilers, M. Garzotto, M. J. Egorin, B. A. Lowe, and W. D. Henner, "Weekly high-dose calcitriol and docetaxel in metastatic androgen-independent prostate cancer," *J. Clin. Oncol.,* vol. 21, pp.123–128, Jan. 2003.

[83] G. G. Schwartz and B. S. Hulka, "Is vitamin D deficiency a risk factor for prostate cancer? (Hypothesis)," *Anticancer Res.*, vol. 10, pp. 1307–1311, Sept.–Oct. 1990.

[84] P. Tuohimaa, L. Tenkanen, M. Ahonen, S. Lumme, E. Jellum, G. Hallmans, P. Stattin, S. Harvei, T. Hakulinen, T. Luostarinen, J. Dillner, M. Lehtinen, and M. Hakama, "Both high and low levels of blood vitamin D are associated with a higher prostate cancer risk: A longitudinal, nested case-control study in the Nordic countries," *Int. J. Cancer*, vol. 108, pp. 104–108, Jan. 2004.

[85] J. Ahn, U. Peters, D. Albanes, M. P. Purdue, C. C. Abnet, N. Chatterjee, R. L. Horst, B. W. Hollis, W. Y. Huang, J. M. Shikany, and R. B. Hayes, "Serum vitamin D concentration and prostate cancer risk: a nested case-control study," *J. Natl. Cancer Inst*., vol. 100, pp. 796–804, June 2008.

[86] TopBraid Composer Website [Online]. Available: http://www.franz.com/agraph/tbc/. Accessed: 08/11/2009.

[87] Franz Inc. Official Website [Online]. Available: http://www.franz.com/. Accessed: 08/11/2009.

[88] M. Padgett. (2008, April 25). Cmap tools for concept mapping and OWL authoring [Online]. Available: http://www.mikepadgett.com/technology/technical/cmaptools-for-concept-mapping-and-owl-authoring. Accessed: 08/11/2009.

[89] V. Gurupur, U. J. Tanik, and M. M. Tanik, "Abstract software design framework: An artificial intelligence-based system for software systems," in *Proc. 20th MAICS Conf.,* Fort Wayne, IN, 2009, pp. 20–25.

[90] V. Gurupur and M. M. Tanik, “Abstract software design framework: A semantic service composition approach,” in *Proc. IEEE Southeast Conf.,* Atlanta, GA, 2009, pp. 295-300.

APPENDIX

BUILDING THE GENE LINKAGE APPLICATION BY USING THE PAAC FRAMEWORK

In this appendix, we provide information on the Gene Linkage Application built by using the PAAC Framework. The application is built with the use of the Apache Tomcat server, which is a freely downloadable web server software. The application begins with the index page, where the user is required to enter the keyword to be searched by the application. This page is illustrated in Figure A.1 of the appendix.

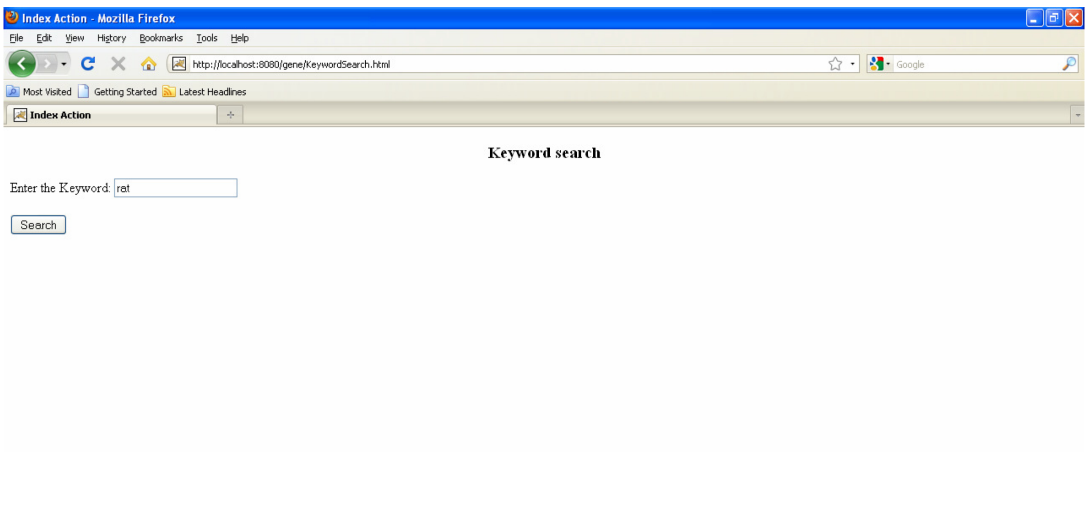


Figure A.1. Keyword search page

This page is followed by the result page, which displays the necessary semantic gene linkage information associated with the keyword. A part of this result is indicated in Figure A.2.

Keyword Search Results

For the Keyword: rat
rat connected_with PubMedID: 15489334
PubMedID: 15489334 connected_with GeneID: 619491
PubMedID: 15489334 connected_with OMIMID: 601665
U58 connected_with ChromosomeNumber: 18
U58 belongs_to GeneID: 619491

rat connected_with PubMedID: 12477932
PubMedID: 12477932 connected_with GeneID: 100129143
PubMedID: 12477932 connected_with OMIMID: 601665
LOC100129143 connected_with ChromosomeNumber: 18
LOC100129143 belongs_to GeneID: 100129143

rat connected_with PubMedID: 16380380
PubMedID: 16380380 connected_with GeneID: 100129143
PubMedID: 16380380 connected_with OMIMID: 601665
LOC100129143 connected_with ChromosomeNumber: 18
LOC100129143 belongs_to GeneID: 100129143

rat connected_with PubMedID: 16335952
PubMedID: 16335952 connected_with no GeneID
PubMedID: 16335952 connected_with OMIMID: 601665
TRNAK-CUU connected_with ChromosomeNumber: 18
TRNAK-CUU belongs_to no GeneID

rat connected_with PubMedID: 15588985
PubMedID: 15588985 connected_with no GeneID
PubMedID: 15588985 connected_with OMIMID: 601665
TRNAK-CUU connected_with ChromosomeNumber: 18
TRNAK-CUU belongs_to no GeneID

Figure A.2 Result of the keyword search

This result is displayed on the basis of the concept map indicated in Figure 19 of the dissertation. This concept map is converted into an ontology file (see Figure 21 of the dissertation). This ontology file is used to extract the RDF triples indicated in Figure 22 of the dissertation. The Java Server Page code to generate this result is as indicated here:

```
<html>

<%@ page import = "java.sql.*" %>

<%@ page import = "java.util.*" %>

<%@ page session = "true" %>
```

```
<head>
<title> Index Action </title>

</head>
<body>
<center><h3>Keyword Search Results</h3></center>

<%
/* Author: Varadraj Prabhu Gurupur
   Date: November 2009
*/
class genepubmed
 {
  int Field4;
  int Field3;
  int Field2;
  int GeneEntryNumber;
  int flag = 0;
  int OmimField3;
  String[] subject = new String[10];
  String[] predicate = new String[10];
  String[] object = new String[10];
  int tripleCount = 0;
  String DataKeyword;
  String DataAbstract;
  int DataPubMedID;
  int DataChromosomeNumber;
  float DataBuildVersion;
```

```
 float DataOMIMID;
  String DataGeneName;
  int DataGeneID;
  String[] subjectFound = new String[10];
  String[] objectFound = new String[10];
  String[] predicateFound = new String[10];
  String[] subjectFoundType = new String[10];
  String[] objectFoundType = new String[10];

  int SubjectFoundFlag = 0;
  int keywordFlag = 0;
  String[] parameterList = new String[10];
  String[] parameterType = new String[10];
  int m;
  int pl;
  int obj;
   }

  // Connecting to the Genelist database
  try
  {
  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

  Connection conn = DriverManager.getConnection("jdbc:odbc:GeneList","
","" ");

  Statement stmt = conn.createStatement();
  Statement stmt4 = conn.createStatement();
  Statement stmt5 = conn.createStatement();
```

```
  Statement stmt6 = conn.createStatement();
  Statement stmt7 = conn.createStatement();
  Statement stmt8 = conn.createStatement();

  genepubmed genepub = new genepubmed();

 ResultSet rs =   stmt.executeQuery("Select * from RDFTriples");

  while(rs.next())
  {
   genepub.subject[genepub.tripleCount] = rs.getString("subject");
   genepub.predicate[genepub.tripleCount] = rs.getString("predicate");
   genepub.object[genepub.tripleCount] = rs.getString("object");
   genepub.tripleCount ++;
  }

  ResultSet rs11 = stmt7.executeQuery("Select * from ParameterList");

  while(rs11.next())
  {
   genepub.parameterList[genepub.pl] =
rs11.getString("ParameterContent");
   genepub.parameterType[genepub.pl] = rs11.getString("ParameterType");
   genepub.pl++;
   }

   for(int k = 0; k <genepub.tripleCount; k++)
   {
         for(int r = 0; r < genepub.pl; r++)
```

```
{

if(genepub.subject[k].equals(genepub.parameterList[r]))
{
    genepub.subjectFound[genepub.m] = genepub.parameterList[r];
 genepub.subjectFoundType[genepub.m] = genepub.parameterType[r];
  genepub.predicateFound[genepub.m] = genepub.predicate[k];

  genepub.m++;
 out.println("<br>");
 }
    }
  }
for(int a = 0; a <genepub.tripleCount; a++)
{
  for(int b = 0; b < genepub.pl; b++)
 {
  if(genepub.object[a].equals(genepub.parameterList[b]))
  {
    genepub.objectFound[genepub.obj] = genepub.parameterList[b];
 genepub.objectFoundType[genepub.obj] = genepub.parameterType[b];
  genepub.obj++;
 out.println("<br>");
 }
   }
    }

stmt.close();
Statement stmt2 = conn.createStatement();
```

```
String Keyword = request.getParameter("Keyword");
genepub.DataKeyword = Keyword;

ResultSet rs1 = stmt8.executeQuery("Select * from pubmedresults");
Statement stmt3 = conn.createStatement();

 while(rs1.next())
 {
 int PID = rs1.getInt("PID");
 String Abstract = rs1.getString("Abstract");
 int field4 = rs1.getInt("Field4");

 if(Abstract.contains(Keyword.trim()))
 {
 genepub.DataAbstract = Abstract;
 genepub.DataPubMedID = PID;
 out.println("<br>");
 out.println("<br><br>");
 out.println("<br><br><br>");
 out.println("<br>");
 out.println("<br><br>");
 genepub.flag = 1;
 genepub.Field4 = field4;
 String f4 = Integer.toString(field4);

ResultSet rs2 = stmt2.executeQuery("Select * from genepubmed1");
while(rs2.next())
{
 genepub.Field3 = rs2.getInt("Field3");
```

```
if(genepub.Field3 == genepub.Field4)
{
 genepub.Field2 = rs2.getInt("Field2");
 }
}

ResultSet rs3 = stmt2.executeQuery("Select * from genelist");
while(rs3.next())
{
 int EntryNo = rs3.getInt("EntryNumber");
 if(EntryNo == genepub.Field2)
 {
  String GeneName = rs3.getString("GeneName");
  genepub.DataGeneName = GeneName;
  int GeneID = rs3.getInt("GeneID");
  genepub.DataGeneID = GeneID;
  int ChromosomeNumber = rs3.getInt("ChromosomeNumber");
  genepub.DataChromosomeNumber = ChromosomeNumber;
  float BuildVersion = rs3.getFloat("BuildVersion");
  genepub.DataBuildVersion = BuildVersion;
  genepub.GeneEntryNumber = EntryNo;
  out.println("<br>");
      }
 }

 ResultSet rs4 = stmt2.executeQuery("Select * from geneomim");
 while(rs4.next())
 {
  int field2 = rs4.getInt("Field2");
```

```
      if(genepub.GeneEntryNumber == field2)
      {
       int field3 = rs4.getInt("Field3");
       genepub.OmimField3 = field3;
       }
      }
    ResultSet rs5 = stmt2.executeQuery("Select * from omimresults");
       while(rs5.next())
      {
        int field4Omim = rs5.getInt("Field4");
        if(genepub.OmimField3 == field4Omim)
        {
         float OmimID = rs5.getFloat("OMIMID");
         genepub.DataOMIMID = OmimID;


                   }
        }


  stmt3.executeUpdate("INSERT INTO Test VALUES('" + genepub.DataKeyword
+ "', '" + genepub.DataGeneID + "', '" + genepub.DataGeneName + "', '"
+ genepub.DataOMIMID + "', '" + genepub.DataPubMedID + "', '" +
genepub.DataChromosomeNumber + "', '" + genepub.DataBuildVersion +
"')");


  for(int n = 0; n < genepub.m; n++)
       {
        ResultSet rs10 = stmt5.executeQuery("Select * from Test");
        while(rs10.next())
        {
```

```
 String key = rs10.getString("Keyword");
if(genepub.DataKeyword.equals(key))
{
 if(genepub.keywordFlag == 0)
 {
 out.println("For the Keyword: " + key);
 out.println("<br>");
 genepub.keywordFlag = 1;
 }
if(genepub.subjectFoundType[n].contains("Integer"))
{
 int SubjectFoundInteger = rs10.getInt(genepub.subjectFound[n]);
 if(SubjectFoundInteger != 0)
{
 genepub.SubjectFoundFlag = 1;
 out.println(genepub.subjectFound[n] + ": " + SubjectFoundInte-
ger);
 out.println(genepub.predicateFound[n]);
     }
 }
if(genepub.subjectFoundType[n].contains("Text"))
 {
 if(genepub.subjectFound[n].contains("Keyword"))
{
genepub.SubjectFoundFlag = 1;
out.println(key);
out.println(genepub.predicateFound[n]);
}
 else
```

```
        {
        String SubjectFoundString =
rs10.getString(genepub.subjectFound[n]);
        genepub.SubjectFoundFlag = 1;
        out.println(SubjectFoundString);
        out.println(genepub.predicateFound[n]);
        }
        }
        if(genepub.subjectFoundType[n].contains("Double"))
        {
        float SubjectFoundFloat =
rs10.getFloat(genepub.subjectFound[n]);
        if(SubjectFoundFloat != 0)
        {
        genepub.SubjectFoundFlag = 1;
        out.println(genepub.subjectFound[n] + ": " + SubjectFoundFloat);
        out.println(genepub.predicateFound[n]);
        }
        }
        if(genepub.objectFoundType[n].contains("Integer"))
       {
        int ObjectFoundInteger = rs10.getInt(genepub.objectFound[n]);
        if((ObjectFoundInteger != 0) && (genepub.SubjectFoundFlag == 1))
        out.println(genepub.objectFound[n] + ": " + ObjectFoundInteger);
        if((ObjectFoundInteger == 0) && (genepub.SubjectFoundFlag == 1))
                    out.println("no " + genepub.objectFound[n]);
        }

        if(genepub.objectFoundType[n].contains("Text"))
```

```
{
if(genepub.objectFound[n].contains("Keyword"))
{

out.println(key);
}
else
{
String ObjectFoundString =
rs10.getString(genepub.objectFound[n]);
out.println(genepub.objectFound[n] + ": " + ObjectFoundString);
}
}
if(genepub.objectFoundType[n].contains("Double"))
{
float ObjectFoundFloat = rs10.getFloat(genepub.objectFound[n]);
if((ObjectFoundFloat != 0) && (genepub.SubjectFoundFlag == 1))
out.println(genepub.objectFound[n] + ": " + ObjectFoundFloat);
}

genepub.SubjectFoundFlag = 0;
out.println("<br>");
}
}
  }
     stmt6.executeUpdate("Delete * from Test");
        }
  }
```

```
    }// end try
  catch(Exception e)
  {
   out.println(e);
    }
           %>
</body>
</html>
```