

# **Organization module of Vijjana, a Pragmatic Model for Collaborative, Self-organizing, Domain Centric Knowledge Networks**

**Gaurav Sudhakar Narkhede**

**Problem Report submitted to the  
College of Engineering and Mineral Resources  
at West Virginia University  
in partial fulfillment of the requirements  
for the degree of  
Master of Science  
in  
Electrical Engineering**

**Approved by**

**Dr. Yenumula. V. Reddy, Ph.D., Chair**

**Dr. James D Mooney, Ph.D.**

**Dr. Sumitra Reddy, Ph.D.**

**Lane Department of Computer Science and Electrical Engineering**

**Morgantown, West Virginia**

**2008**

**Keywords: Organization, Vijjana, ASP.NET, VB.NET**

# Abstract

---

## **Organization module of Vijjana, a Pragmatic Model for Collaborative, Self-organizing, Domain Centric Knowledge Networks**

**Gaurav Sudhakar Narkhede**

Currently the main Navigation System of World Wide Web is Search Engines like Google, Yahoo etc. Search Engines harvest billions of Web Pages out there residing on servers with the help of computer robot programs called Spiders or Crawlers. These Spiders crawl the World Wide Web for web pages and stores them into the Search Engine's database. Search Engine searches this database upon queried based on Keywords or Phrases. Keyword based searching has some inherent limitations because of which user has to go through number of irrelevant web pages before finding the desired information. Transforming current web into a semantic web is the ultimate solution; it is highly unlikely to be materialized in the near future. Another possible solution is Social Bookmarking websites that stores and organizes web pages on the internet with the help of metadata entered by users of these sites. However with the growth in number of pages, search process becomes unmanageable again.

To address this problem we have developed Vijjana, a model for Collaborative, Self Organizing, and Domain Centric Knowledge Networks. Vijjana is an agent based model in which multiple users can submit web pages to an agent which establishes a relationship between pages based on metadata by using predefined taxonomy and semantic structure. The model also provides other mechanisms which are carried out by different agents such as rating, visualization, self organization etc.

In this report we will explain the implementation of an Organizing Agent for Vijjana Framework. The Organizing Agent is a component in the Vijjana which is responsible for organizing a Jan into the evolving Vijjana, recording information about Jan in a Provenance File, automatic notifications and responding to the events generated by the Visualization Agent.

## Acknowledgments

I gratefully acknowledge Committee Chair, Dr. Yenumula V. Reddy and am very thankful for his supervision, constant support, and proper guidance throughout the project. He always encourages me to come up with new ideas and helped me a lot in implementing those ideas. This project is originated from his ideas and evolved from the insights and suggestions of Vijjana team.

I would like to extend my thanks to Dr. James Mooney and Dr. Sumitra Reddy for serving on the committee and supporting the project with their suggestions and encouragement.

I would like to thank all my friends and project mates who helped me with their suggestions and always being supportive in completing this project.

Finally, I am deeply thankful to my beloved parents and sister for their encouragement and support in all the things I do.

# Table of Contents

---

Abstract .....	ii
Acknowledgments .....	iii
<b>Chapter 1 Introduction</b> .....	<b>1</b>
1.1 Background .....	1
1.1.1 What is the current situation of the Web? .....	1
1.1.2 What are problems with current approach? .....	1
1.1.3 What are available solutions? .....	2
1.1.4 What is Vijjana? .....	4
1.2 Problem statement .....	5
<b>Chapter 2 Vijjana Framework</b> .....	<b>6</b>
2.1 Introduction .....	6
2.2 Vijjana Agents .....	7
2.2.1 Taxonomy and Semantic Net of Knowledge (T and R) .....	7
2.2.2 The Discovery Agent (dA) .....	7
2.2.3 The Organizing Agent (oA) .....	7
2.2.4 The Consistency/Completeness Agent (cA) .....	8
2.2.5 The Visualization Agent (vA) .....	8
2.2.6 The Search Agent (sA) .....	9
2.2.7 The Rating Agent (rA) .....	9
2.3 Vijjana Framework Architecture .....	10
2.3.1 Administrator Interface .....	11
2.3.2 General User Interface .....	11
2.3.3 Social Network Management .....	11
2.3.4 Jan Discovery .....	12
2.3.5 Database Server .....	12
2.4 Vijjana Client Architecture .....	13
2.5 Vijjana Database .....	14
<b>Chapter 3 Technologies Used</b> .....	<b>15</b>
3.1 .NET Framework 3.5 .....	15

3.1.1 Common Language Runtime (CLR) .....	15
3.1.2 .NET Framework Class Library.....	18
3.1.3 Advantages of .NET Framework.....	18
3.2 ASP.NET.....	19
3.3 ADO.NET.....	20
3.4 VB.NET 9.0.....	21
3.5 AJAX.....	21
3.6 AJAX Control Toolkit Framework .....	22
3.7 Cascading Style Sheets (CSS).....	22
<b>Chapter 4 Organizing Agent</b> .....	<b>23</b>
4.1 What is an Organizing Agent?.....	23
4.2 Functions of an Organizing Agent .....	23
4.2.1 Organization of Jan .....	23
4.2.2 Maintaining Provenance File .....	24
4.2.3 Automatic Notifications about incorporation of new Jan .....	24
4.2.4 Respond to the events generated by Visualization Agent.....	25
4.3 Architecture of an Organizing Agent .....	25
4.3.1 Advantages of Layered Architecture.....	26
4.3.2 Data Layer .....	27
4.3.3 Data Access Layer.....	28
4.3.4 Business Logic Layer.....	29
4.3.5 Presentation Layer .....	30
<b>Chapter 5 VIJANA Namespace</b> .....	<b>31</b>
5.1 DatabaseManipulation.....	31
5.2 Entities .....	32
5.3 DataAccessLayer .....	32
5.4 BusinessLogicLayer.....	33
5.5 ConsistencyAgent.....	34
5.6 Notifications.....	35
5.7 Enumerations .....	35
5.8 SearchServices .....	36
5.9 Email.....	36

<b>Chapter 6 Implementation</b> .....	37
6.1 New User Registration .....	37
6.2 Login .....	40
6.3 Edit Account Information.....	42
6.4 Password Recovery .....	44
6.5 Markup.....	45
6.6 Search.....	47
6.7 Add Keywords, Rename Keywords or Remove Keywords .....	49
6.8 Keyword Subscription .....	50
<b>Chapter 7 Future Work and Conclusion</b> .....	52
7.1 Future Work .....	52
7.2 Conclusion.....	52
<b>References</b> .....	54
Appendix A: Presentation Slides .....	55

# Chapter 1 Introduction

---

## 1.1 Background

### 1.1.1 What is the current situation of the Web?

Currently the main Navigation System of World Wide Web is Search Engines like Google, Yahoo etc. Search Engines harvest billions of Web Pages out there residing on servers with the help of computer robot programs called Spiders or Crawlers. These Spiders crawl the World Wide Web by following the links in the pages they already have in their database. After Spiders find pages, they are indexed using computer program called Indexer which identifies text, links and other content of the page and stores it into the Search Engine's database. Search Engine searches this database upon queried based on Keywords or Phrases. So whenever we search the World Wide Web using Search Engines we are actually searching somewhat old copy of the real web page.

### 1.1.2 What are problems with current approach?

Keyword based searching has some inherent limitations because of which user has to go through number of irrelevant web pages before finding the desired information.

Limitations of Keyword based searching-

- ✓ Constrained by Keyword or Phrase based matching techniques which results in low precision and recall.
- ✓ User frequently fails to describe the information he/she is searching for in few words.
- ✓ User must be intelligent.
- ✓ Synonym terms are not taken into consideration.
- ✓ Do not retrieve conceptual terms.
- ✓ Semantic Similarity and Semantic Relatedness are ignored.

### **1.1.3 What are available solutions?**

#### 1. Transform current web into a semantic web

What is semantic web?

"If HTML and the Web made all the online documents look like one huge **book**, RDF, schema, and inference languages will make all the data in the world look like one huge **database**" *Tim Berners-Lee, Weaving the Web, 1999*

The word semantic stands for *meaning of* and the term Semantic Web means describing things in a way that computer applications can understand.

Problems with this solution-



- ✓ Semantic Web uses RDF (Resource Description Framework) as a markup language for describing information and resources on the web. RDF was developed by people with academic background in artificial intelligence so it has a steep learning curve as per as traditional developers are concerned.
- ✓ It is highly unlikely to materialize in the near future.

## 2. Social Bookmarking Sites

What are Social Bookmarking Sites?

Social Bookmarking sites stores and organizes web pages on the internet with the help of metadata entered by users of these sites. Social Bookmarking sites allow users to store and manage web pages by tagging then with keywords. This allows easy discovery of related web pages later.

Problems with this solution-

- ✓ Lack of controlled vocabulary for tagging web pages
- ✓ Incorrect tagging because of spelling mistakes, singular vs. plural, capitalization etc
- ✓ Tags with several meanings
- ✓ No way of indicating hierarchical relationships between tags
- ✓ With the growth in number of pages, search process becomes unmanageable again.

### **1.1.4 What is Vijjana?**

Vijjana, a Sanskrit word that means knowledge obtained through classification and analysis, is a model for Collaborative, Self Organizing, and Domain Centric Knowledge Networks. Vijjana is an agent based model in which multiple users can submit web pages to an agent which establishes a relationship between pages based on metadata by using predefined taxonomy and semantic structure. The model also provides other mechanisms which are carried out by different agents such as rating, visualization, self organization etc.

Advantages of Vijjana-

- ✓ Offers controlled vocabulary for tagging web pages
- ✓ Establishes hierarchical relationship between tags
- ✓ Because of domain centric nature of the Vijjana model, it is easier to manage and classify growing number of web pages
- ✓ Semantic Similarity and Semantic Relatedness are taken into consideration
- ✓ Leverages the power of existing Knowledge Base such as Open Directory Project database

## **1.2 Problem statement**

To develop an Organizing Agent for Vijjana Framework which will automatically classify, organize and interlink Jans (URIs) into the evolving Vijjana Framework based on the markup information entered by the user.

# Chapter 2 Vijjana Framework

---

## 2.1 Introduction

Vijjana is an agent based framework and we define Vijjana as follows-

$$\text{Vijjana-X} = \{J, T, R, dA, oA, cA, vA, sA, rA\}$$

Where

X = the domain name

J= the collection of JAN's in the Vijjana-X

T = the Taxonomy used for classification of JAN's

R= the domain specific relations

dA = the discovery agent which find relevant JAN's

oA = the organizing agent which interlinks the JAN's based on R

cA = the consistency/completeness agent

vA = the visualization agent

sA = the search agent

rA = the rating agent

Each agent is described in brief in following sections.

## **2.2 Vijjana Agents**

### **2.2.1 Taxonomy and Semantic Net of Knowledge (T and R)**

The Taxonomy is the practice and science of classification. As Vijjana is a domain centric knowledge network, it requires us to define appropriate Taxonomy for domain under consideration. Vijjana treats Taxonomies as parameters which allow it to work with any Taxonomy and associated related semantics. This allows Vijjana framework to import and export knowledge bases as long as there is agreement on an ontology, taxonomy and related semantics.

### **2.2.2 The Discovery Agent (dA)**

In current version of Vijjana the Discovery Agent is a Human (user). The user discovers a Jan through normal search process, marks up that Jan and submit it to Vijjana server where it is classified and organize into appropriate knowledge network depending on Jan markup information.

### **2.2.3 The Organizing Agent (oA)**

Once the Jan is submitted to Vijjana server by Discovery Agent through Vijjana Client Interface it is handed off to the Organizing Agent which

- ✓ Ensures that the Jan represents the genuine link
- ✓ Classify and interlink the Jan depending on the Jan markup information

- ✓ Logs the information in the provenance file which is later used for Jan history
- ✓ Notify users about the incorporation of new Jan.

#### **2.2.4 The Consistency/Completeness Agent (cA)**

The Consistency/Completeness Agent is responsible for maintaining the integrity of Vijjana Knowledge Network. The Consistency/Completeness Agent periodically visits URI's of the Jans to make sure they exist. The URL of Jan is considered as alive by Consistency/Completeness Agent if it has 200 (OK) HTTP status code and it is considered as broken if it has 404 (Not found) HTTP status code.

#### **2.2.5 The Visualization Agent (vA)**

The Visualization Agent is responsible for creating and displaying different views of Vijjana Knowledge Network based on user preference. Vijjana currently supports three views-

- ✓ Hyper Tree View
- ✓ Radial Graph View
- ✓ Hyper Graph View
- ✓ Tree View

### **2.2.6 The Search Agent (sA)**

The Search Agent provides functionality for searching Vijjana Database.

Search Agent implements variety of search mechanisms such as –

- ✓ Keyword Based Search
- ✓ Phrase Based Search
- ✓ Search based on Related Semantics
- ✓ Search by Category, Jan rating, date etc

### **2.2.7 The Rating Agent (rA)**

The Rating Agent inserts the rating of Jan submitted by user into Vijjana Database. The Rating Agent is also responsible for determining high value Jans based on various factors such as average Jan rating, user defined values, number of hits etc

## 2.3 Vijjana Framework Architecture

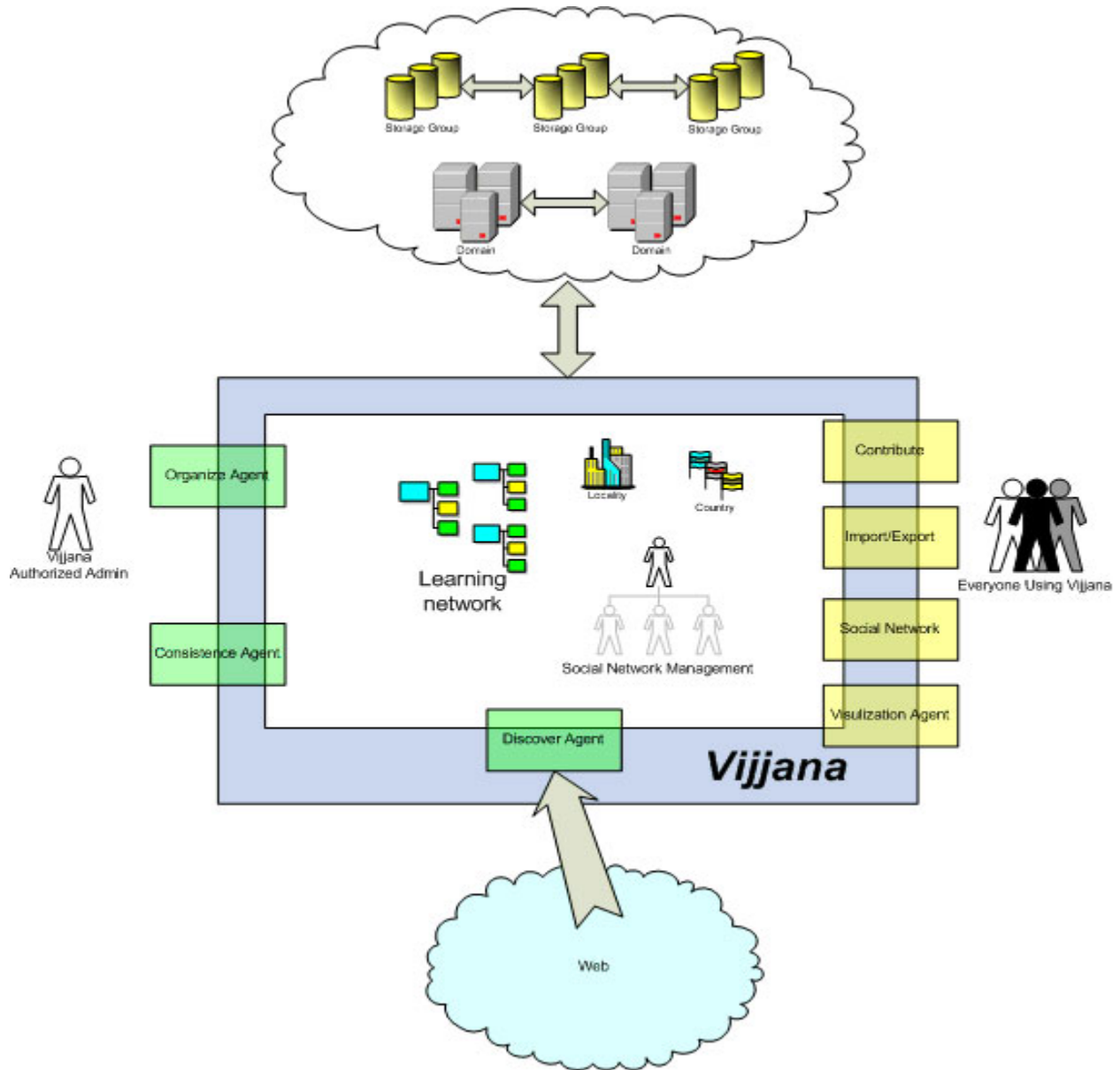


Figure 1: The Vijjana Architecture

Figure shows the high level view of the Vijjana Architecture with different components of Vijjana such as Administrator Interface, General User



Interface, Social Network Management, Jan Discovery and Vijjana Database Server. Each of these components is explained in brief in following sections.

### **2.3.1 Administrator Interface**

Administrator Interface allows Administrator of Vijjana to monitor Vijjana Database Server, Approve/Disapprove Content, manage SMTP server, and run Jan consistency check against the Vijjana Database.

### **2.3.2 General User Interface**

General User Interface allows end user to contribute Jans to Vijjana using dedicated client interface or using browser add-ons. It also allows end user to import/export entire knowledge networks and visualize Vijjana using visualization agent.

### **2.3.3 Social Network Management**

Vijjana Social Network makes user aware of other users interacting with the knowledge network and allows them to open a communication channel between each other. This may result in achieving certain task such as rating or reviewing Jan, commenting on a Jan or classifying a Jan collaboratively.

### **2.3.4 Jan Discovery**

Jan Discovery allows the Jan to migrate from web to Vijjana Knowledge Network.

### **2.3.5 Database Server**

The database server contains the taxonomies, link relationships and the actual URIs (uniform Resource Identifiers) representing the Jans in individual knowledge networks.

## 2.4 Vijjana Client Architecture

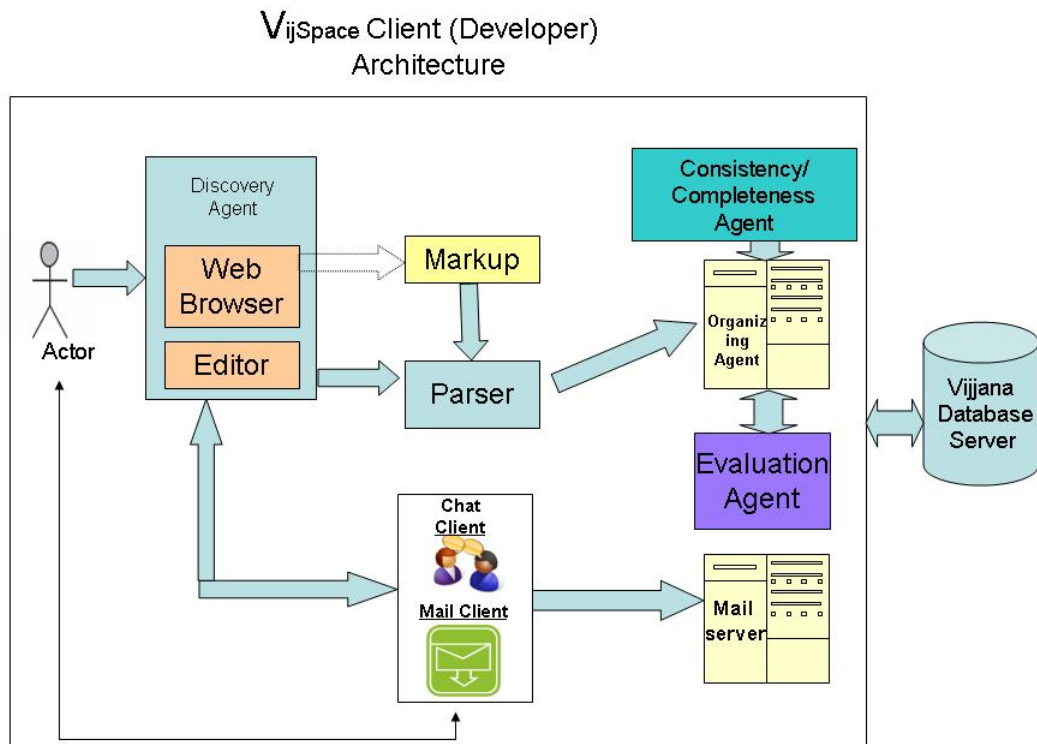


Figure 2: Vijjana Client Architecture

The user can submit URI's to Vijjana Server through one of the two following ways-

- ✓ Dedicated Vijjana Client Interface
- ✓ Vijjana Browser Extensions for Internet Explorer and Firefox

Whenever the Jan (URI) arrives to Vijjana Server, it is parsed to extract the markup information provided by user to organize the Jan in to Vijjana

Knowledge Network. In later implementations of Vijjana we plan to allow user to submit URI's to Vijjana Server using

- ✓ Email
- ✓ RSS

## **2.5 Vijjana Database**

The Vijjana Database consists of three main parts basic raw data, user data and website data. The basic raw data is gathered form Open Directory Project (ODP) RDF Dump. This RDF Dump is converted into MySQL Database. The ODP data act as directory standard for World Wide Web and has been widely used in many areas, providing data for many big search engines such as Google, Yahoo etc. The user database is used to store user information and his recent activities. Keeping track of user history is necessary to find out user interest which will help suggesting more relevant results to user in future. The website database is used by Vijjana website for its internal data storage.

# Chapter 3 Technologies Used

---

## 3.1 .NET Framework 3.5

The Microsoft .NET Framework is a platform for building, deploying, and running Web Services and applications. It provides a highly productive, standards-based, multi-language environment for integrating existing investments with next-generation applications and services as well as the agility to solve the challenges of deployment and operation of Internet-scale applications. The .NET Framework consists of two main parts: the common language runtime, and hierarchical set of unified class libraries.

### 3.1.1 Common Language Runtime (CLR)

Common Language Runtime is Microsoft's implementation of Common Language Infrastructure (CLI). CLI is an open specification provided by Microsoft which defines an environment that allows writing programs in multiple high level languages in architecture and platform independent way. CLI has four main aspects-

## **Common Type System (CTS)**

All the programming languages in .NET framework share a common set of types and operations which are known as Common Type System.

## **Metadata**

Metadata is data about program structure. The program structure remains same across multiple programming languages. Common Language Runtime (CLR) uses the metadata to understand the code written in different .NET compliant languages and to combine them.

## **Common Language Specification (CLS)**

Any language that wants to use Common Language Infrastructure (CLI) must conform to rules defined by Common Language Specification. All the Common Language Specification (CLS) compliant languages are interoperable i.e. the code written in C# can be used in VB.NET or any other .NET language.

## **Virtual Execution System (VES)**

Virtual Execution System is the Virtual Machine used by Common Language Runtime (CLR) to execute the Common Language Infrastructure (CLI) compliant programs. There are different Virtual Execution Systems (VES) for different operating systems which makes .NET a platform independent framework.

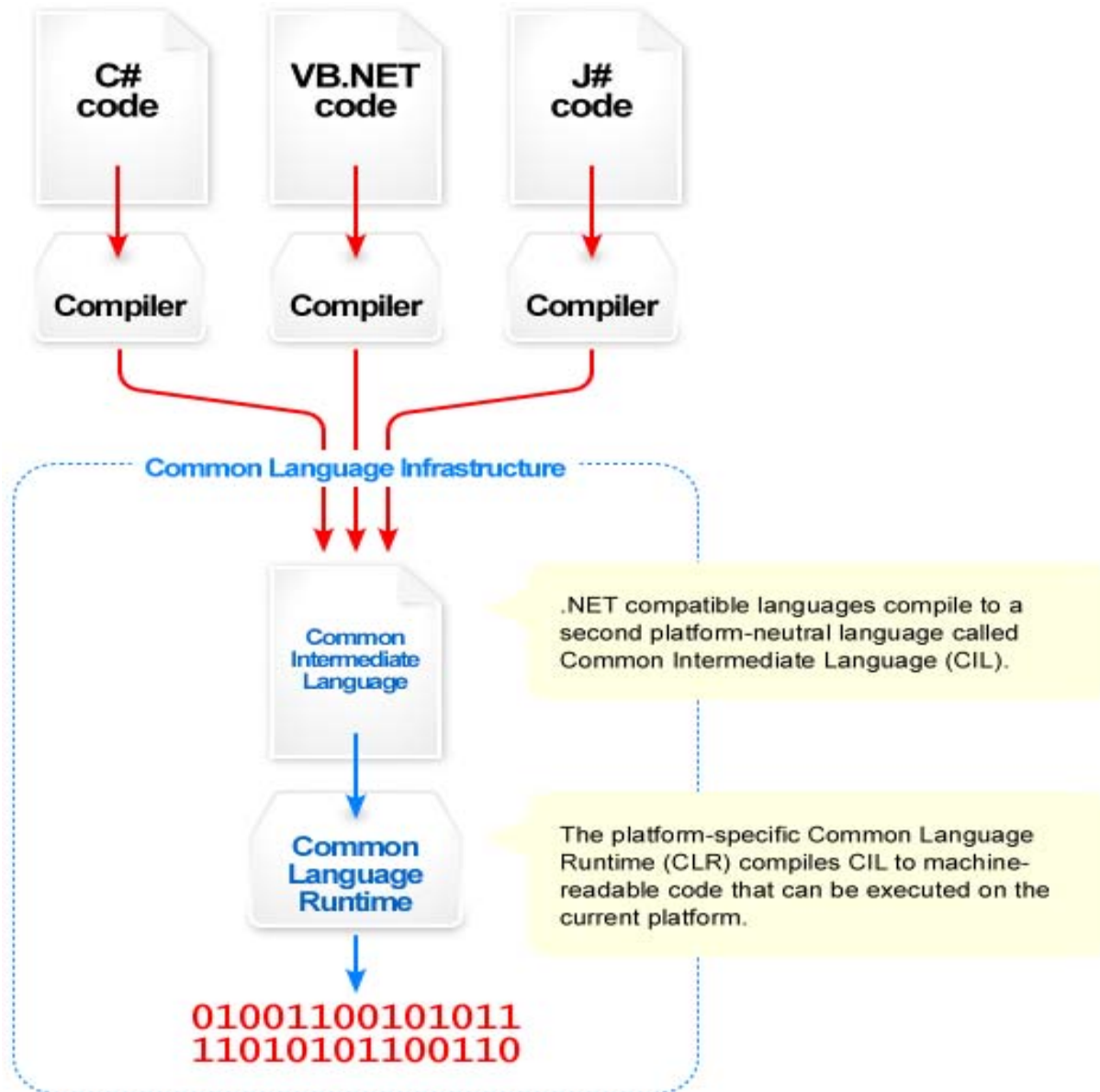


Figure 3: Visual overview of the Common Language Infrastructure (CLI)

(Wikipedia)

.NET Framework comes with command line compilers for each language supported by the framework. The code written in higher level languages such as C#, VB.NET, J# or Cobol.NET is converted into Common Intermediate Language (CIL) or Microsoft Standard Implementation

Language (MSIL) by respective language compilers. Once the code is converted into CIL it is stored into assemblies. Then the platform specific Common Language Runtime (CLR) picks up the CIL code from assembly and convert it into machine level language specific to particular platform. Common Language Runtime (CLR) does not compile everything in the assembly; it only compiles the method that is being invoked. This technique is referred to as jitting which happens once per method call. The next time this method is called Common Language Runtime (CLR) uses previously complied code.

### **3.1.2 .NET Framework Class Library**

.NET Framework Class Library consists of more than 13,000 classes, interfaces and value types which provide solutions to common programming tasks. These classes are grouped into several logically related sets or API's such as ASP.NET, ADO.NET, and Windows Forms etc.

### **3.1.3 Advantages of .NET Framework**

- ✓ Language Independence
- ✓ Memory management
- ✓ Consistent Programming Model
- ✓ Simplified Development Efforts
- ✓ Easy application deployment and maintenance



## **3.2 ASP.NET**

ASP stands for Active Server Pages. ASP.NET is a technology for server side scripting built on the .NET Framework. ASP.NET provides an event driven server side programming paradigm where HTML and source code are written together. ASP.NET is pure server side technology. ASP.NET code executes on server and produces the HTML dynamically to send back to the browser.

Web pages in ASP.NET have .aspx extension. Each web page is accompanied by code behind page written in C# or VB.NET. This code behind page encapsulates all the programming logic for the page whereas all the User Interface is contained by the .aspx page.

Whenever an ASP.NET page is requested by the browser the ASP.NET Server (IIS Internet Information Services) executes the code behind page and process the user input if any and generates pure HTML to send back to the browser. The source code is compiled the first time the page is requested. Execution is fast as the Web Server compiles the page the first time it is requested. The server saves the compiled version of the page for use next time the page is requested

### 3.3 ADO.NET

ADO.NET stands for ActiveX Data Objects. ADO.NET is a set of software components which is a part of .NET Framework Base Class Library. ADO.NET allows developer to access data from relational as well as non-relational data sources. ADO.NET mainly consists of two parts Data Provider and Data Sets which cleanly separate data access from data manipulation. Data Providers are used for connecting to a database, executing commands, and retrieving results. Those results are either processed directly, or placed in an ADO.NET DataSet object in order to be exposed to the user in an ad-hoc manner, combined with data from multiple sources, or remoted between tiers.

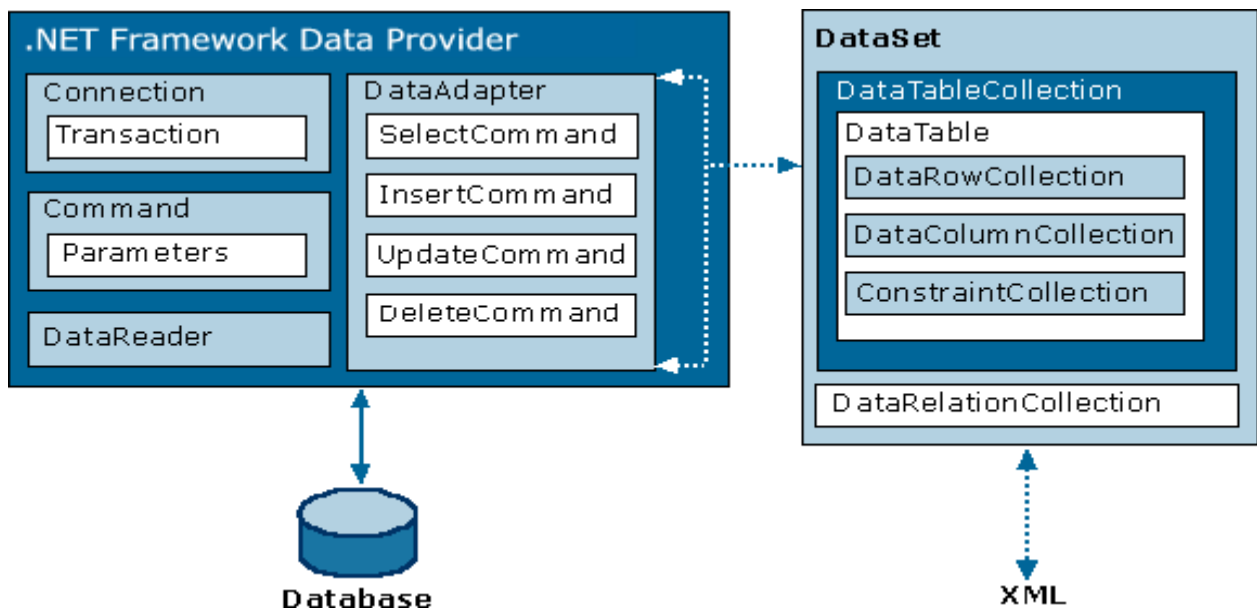


Figure 4: ADO.NET architecture (MSDN Library)

### **3.4 VB.NET 9.0**

VB.NET 9.0 is Common Language Infrastructure compliant and pure Object Oriented Programming Language. It supports all the Object Oriented features such as Inheritance, Encapsulation, Abstraction and polymorphism along with advance features such as reflection, decoupling etc. VB.NET code is compiled as managed code, which means it benefits from the services of the common language runtime. These services include language interoperability, garbage collection, enhanced security, and improved versioning support.

### **3.5 AJAX**

AJAX stands for Asynchronous JavaScript and XML. AJAX is used in web applications to retrieve data from server asynchronously without updating the complete page. This pattern is known as Partial Page Update Pattern. AJAX uses XMLHttpRequest JavaScript object or remote scripting to send and receive data form server. The AJAX technique makes Web Applications smaller, faster and more interactive. AJAX is extensively used in Vijjana Website. Microsoft AJAX Framework 3.5 is used for implementing AJAX functionality in Vijjana Website.

### **3.6 AJAX Control Toolkit Framework**

AJAX Control Toolkit is a set of AJAX enabled ASP.NET software components. AJAX Control Toolkit is open source joint project between the community and Microsoft. AJAX Control toolkit provide many useful widgets such as –

Accordion

Autocomplete

Validator

Popup Control etc

Many of these controls are extensively used in Vijjana Website to provide smooth and more interactive User Interface.

### **3.7 Cascading Style Sheets (CSS)**

Cascading Style Sheets are used to style and layout HTML/XHTML pages. It separates document content from document presentation. This separation makes web pages more accessible to readers as well as to search engine spiders. It also makes the structure of web page more flexible and extendable. CSS Specifications are provided by World Wide Web consortium. All the web pages in Vijjana web site are styled and structured using CSS.

# Chapter 4 Organizing Agent

---

## 4.1 What is an Organizing Agent?

The Organizing Agent is a component in the Vijjana which is responsible for organizing a Jan into the evolving Vijjana, recording information about Jan in a Provenance File, automatic notifications and responding to the events generated by the Visualization Agent. The Organizing Agent acts as a middleware application between Discovery Agent and Visualization Agent.

## 4.2 Functions of an Organizing Agent

The Organizing Agent has four main functions as follows-

- Organization of a Jan
- Maintaining Provenance File
- Automatic Notifications about incorporation of new Jan
- Respond to the events generated by Visualization Agent.

Each of these functions is described in details in following sections.

### 4.2.1 Organization of Jan

Whenever the Discovery Agent discovers a new Jan, it is marked up and submitted to the Organization Agent. After the Organization Agent receives a Jan, it sends the Jan to Consistency/Completeness Agent to ensure that the

Jan represents a non-broken, genuine link. After confirming that the link is genuine the Organizing Agent examines the markup information to classify and interlink the Jan. The Organizing Agent also stores the Jan and attribute values associated with it into a Database so that the Visualization Agent could use this data to display the Vijjana in variety of forms as per user preference.

#### **4.2.2 Maintaining Provenance File**

The Provenance File is used in Vijjana to record the information about the origin of the Jan i.e. the place from where it came. It is the responsibility of Organizing Agent to extract the Provenance information about a particular Jan from the metadata associated with that Jan and record it into the Provenance File. The provenance information will be later used to generate the history of any Jan.

#### **4.2.3 Automatic Notifications about incorporation of new Jan**

One of the features of Vijjana requires that the user should be notified whenever a new Jan (in particular field) is added as per the user preference. It is the responsibility of Organizing Agent to identify such users for every new Jan and notify them after the incorporation of that particular Jan.

#### **4.2.4 Respond to the events generated by Visualization Agent**

Some of the possible events generated by the Visualization Agent are as follows-

- Whenever a user clicks a particular node it is required that the node should be expanded for which a Visualization Agent will require more data. This event will trigger the Organizing Agent and the Organizing Agent will then provide the Visualization Agent with the additional data required.
- Another possible event is when user evaluates any Jan; the resulting value of the evaluation should be entered into a corresponding field in the Database.

#### **4.3 Architecture of an Organizing Agent**

Vijjana is designed to have Layered Software Architecture. It has four main layers Data Layer, Data Access Layer, Business Logic Layer and Presentation Layer. Data Access Layer and Business Logic Layer together provide the functionality of Organizing Agent.

### **4.3.1 Advantages of Layered Architecture**

The N-tier Applications provide specific advantages that are vital to the business continuity of the enterprise. Typical features of a real life n-tier may include the following:

#### **Availability and Scalability**

With n-Tier architecture each layer can be located on physically different servers with only minor code changes; hence they scale out and handle more server load.

#### **Data Abstraction**

The inner implementation of a layer is completely hidden from other layers. This makes it possible to change or update one layer without recompiling or modifying other layer.

#### **Easy Maintenance**

We can add a new layer without redeploying the whole application. For example, by separating data access code from the business logic code, when the database servers change you only needs to change the data access code. Because business logic code stays the same, the business logic code does not need to be modified or recompiled.

Reduces coupling between User Interface, Business Logic and Database.



Layered architecture allows software application development to be more effectively split into teams, with each team working on a different tier simultaneously.

#### **4.3.2 Data Layer**

Data Layer consists of Database. Along with Database Data Layer also includes collection of stored procedures and views. Stored procedures and views allow us to manipulate the data that goes into the database tables or out of the database tables. For example assume that for some reason we have to change the physical structure of the database table. If we reference the database tables directly into the Business Logic then we have to change our code to incorporate the new changes in the database table structure. This is not the case with stored procedures and views. They expose a logical structure to the layer which uses them. If we use stored procedures and views to access data in the database then we can keep the same logical structure by updating stored procedures and views to incorporate the changes in database table structure without modifying our code.

### **4.3.3 Data Access Layer**

Most Web Applications are data driven. Data Driven Web Applications require databases to store data, code to modify and retrieve it and Web Pages to collect and summarize them. In .NET Web Applications ASP.NET Web Pages make up the Presentation Layer of the application. In Data Driven Web Applications one approach to provide data to Presentation Layer is directly embedding data specific logic into ASP.NET code behind pages using ADO.NET. The disadvantage of this approach is that it tightly couples the Data Access Logic to presentation layer. In Vijjana all the Data Access Logic is written in a separate Class Library Project.

The main objective of Data Access Layer is to provide data from database to Business Objects without using the database specific code such as SQL Queries, ADO.NET code etc. To accomplish this Data Access Layer exposes collection of data access methods. These methods operate on Database using database specific code but they do not expose any database specific method parameters or do not return database specific data types. Any time we want to use data from database in Business Logic Layer we call methods from Data Access Layer. This encapsulates all the database specific code into Data Access Layer and makes Business Objects database independent.

Data Access Layer perform some of the very common tasks on the database on behalf of their caller –

- ✓ Create new records in the database
- ✓ Retrieve records from database based on some criteria
- ✓ Update records
- ✓ Delete records

These common operations are also known as CRUD operations. Along with these crud operations Data Access Layer Components can also implement some methods to perform some business logic.

#### **4.3.4 Business Logic Layer**

The Data Access Layer separates Data Access Logic from Presentation Layer but it does not take care of data validation and business rules. The Business Logic Layer encapsulates all the business logic and enforces data validation and business rules. Business Logic Layer is responsible for processing the data retrieved and sent to the presentation layer. The main task of Business layer is business validation and business workflow. For example –

- ✓ For our Vijjana Web Application we want the User Name field of Membership table to 20 characters long and it should only contain Uppercase or Lowercase alphabets. This example comes under validation rules.
- ✓ We might want to disallow unregistered users from accessing certain data. This comes under business rules.

### **4.3.5 Presentation Layer**

Presentation Layer consists of ASP.NET web pages. ASP.NET web page consists of two parts a .aspx file also known as markup file and a .vb file which is known as code behind page. The .aspx file encapsulates all the User Interface for the Web Page and .vb file encapsulate all the logic for User Interface. This approach cleanly separates Designers and Developers. Designers don't have to worry about messing up the code while updating the design and Developers don't have to worry about sifting through User Interface to update code.

# Chapter 5 VIJJANA Namespace

---

Vijjana is implemented a software with layered architecture. Each layer encapsulates some functionality, uses services of lower layer and provides services to the upper layer. Each layer is implemented as a separate namespace contained by VIJJANA namespace. All the namespaces contained by VIJJANA namespace are explained in the following sections.

## 5.1 DatabaseManipulation

This namespace contains only two classes named AdapterFactory and MySQLDatabaseAdapter. AdapterFactory class implements the Abstract Factory Design Pattern. Upon request AdapterFactory class creates an instance of appropriate Database Adapter class. MySQLDatabaseAdapter class is responsible for

- ✓ Connecting to the MySQL Database whenever requested
- ✓ Executing queries and returning the result sets
- ✓ Performing CRUD operations such as SELECT, INSERT UPDATE and DELETE
- ✓ Disposing database specific resources such as Database Adapters, Command Objects etc once the task is done

MySQLDatabaseAdapter is class that encapsulates all the code required to interact with the MySQL Database Server. This approach makes Vijjana

database independent. For example if in the future we decide to change the Database Server from MySQL to Oracle then we only have to add one more class called OracleDatabaseAdapter.

## **5.2 Entities**

The Entities namespace contains all the Business Objects such as Vijjana, User, Keyword, Jan, Group, Folder, Category, Comment, BasicSearchResult, BasicCommentResult etc. These Business Objects represent corresponding tables in the database. These objects also expose several properties corresponding to fields in the database table. These business objects have several overloaded constructors which are suitable for different programming situations.

## **5.3 DataAccessLayer**

The DataAccessLayer namespace contains all the classes which provide functionality of Data Access Layer. These classes contain database specific code to perform different operations on the database and retrieve result sets. Classes in DataAccessLayer namespace return instances of Business Objects as result which removes dependence of a Web Application on database specific data types. Each Business Object in the result set represents a row in database table. This technique is also known as Object

Relational Mapping (ORM). Any operation performed on the object reflects in the corresponding database table row.

## **5.4 BusinessLogicLayer**

The BusinessLogicLayer namespace contains all the classes which provide functionality of Business Logic Layer. These classes enforce data validation and business rules. Classes in Business Logic Layer separate Presentation Layer and Data Access Layer components. Methods in these classes accept data from presentation layer, validate this data against the defined validation rules, apply business rules and insert data into the database. Same sequences of operations are performed in the reverse order while retrieving the data from the database. Classes in BusinessLogicLayer namespace also takes care of Exception Handling, transactions and rollbacks.

### **Example of exception handling**

While performing a certain operation if a method throws an exception then the class will notify the developer if the exception is related to business rules or it will notify the end user if the exception is related to validation rules.

### **Example of Transaction and Rollbacks**

The transaction and rollbacks are used only if the database server does not support these features. Many times we have to insert data into several tables simultaneously for example while marking up a Jan we have to insert data into Jan table, Keyword table, JanRating table etc. While executing a sequence of insert operations if one operation fails and raises the exception then the remaining operations will not be executed. This will result in incomplete record in the database. To maintain the integrity of the data in the database we have to rollback (delete the corresponding entries in the database) the insert operations which were successful.

## **5.5 ConsistencyAgent**

The ConsistencyAgent namespace contains class named URIVValidation that validates URI of the Jan and also periodically checks the HTTP status of the URI. URIVValidation class uses Regular Expressions to check the format of the URI. This ensures that the URI is in valid format before entering it into the database. The URIVValidation class also checks the HTTP status of the URI to check its existence. An URI is considered to be in existence only if it has an HTTP status code equal to 200 (OK). The count of number of times an URI found nonexistent is kept into the database. If this count increases to a certain value say five the URI and its related information is automatically deleted from the database.



## 5.6 Notifications

The Notifications namespace contains two classes namely Notifications and Messages. The Notifications class is responsible for generating custom messages for errors, exceptions and for giving information to user. It wraps up the message into JavaScript alert box which is then displayed by the browser. The Messages class stores all the text for the messages as public strings. This approach enables Vijjana to provide information to the end user in several languages. For example if in the future we want to provide the Vijjana in German language along with English then we only have to add new Messages class that contains string variables containing corresponding messages in German language. This technique is also known as Localization.

## 5.7 Enumerations

The Enumerations namespace contains all the public enum used in Vijjana. Enumerations provide a way of representing numerical values as symbolic text for example for a three star rating system we can define an enumeration as follows

Low = 1

Medium = 2

High = 3

Use Enumerations increases the readability of the code.

## **5.8 SearchServices**

The SearchServices namespace provide the basic framework for search agent. This namespace contains two classes namely BasicSearchResults and BasicCommentResults. These classes get data from Data Access Layer based on search criteria and convert it into more convenient form so that it can be directly used into ASP.NET pages.

## **5.9 Email**

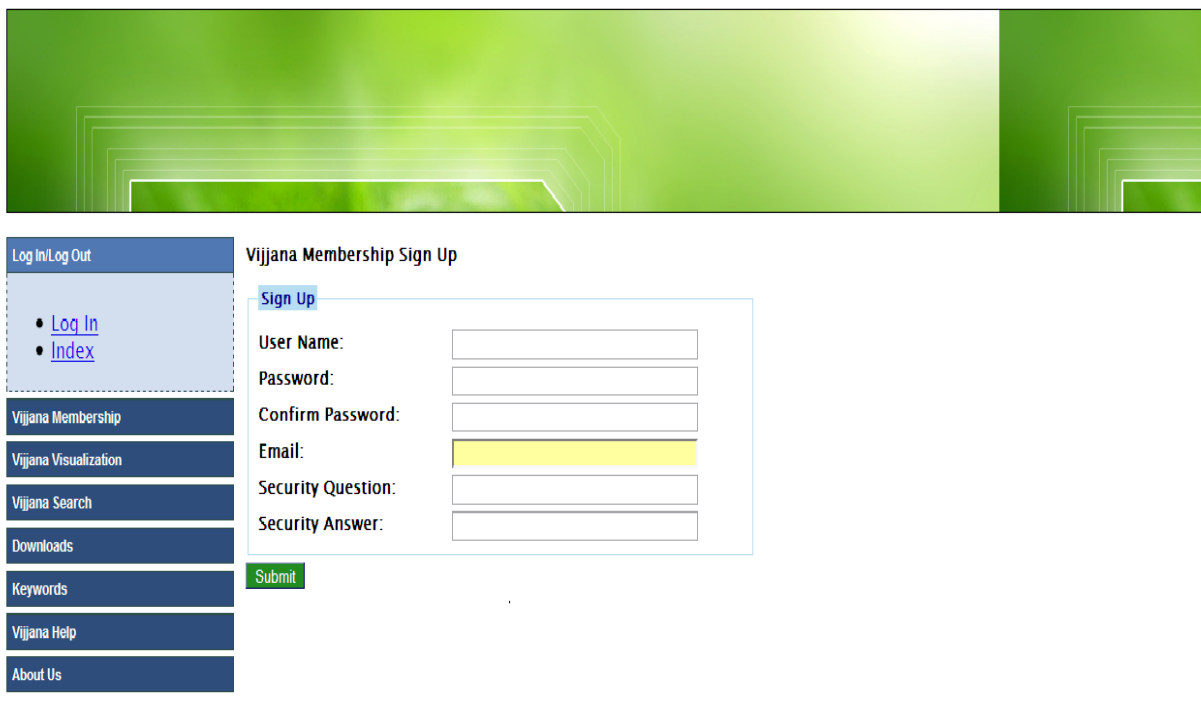
The Email namespace provide the Email functionality for Vijjana. It contains two classes namely VijjanaEmail and EmailTemplates. VijjanaEmail class performs the task of sending emails to users. This class uses Gmail's SMTP server, port 587 and 'Admin.Vijjana@gmail.com' as username. In the future we can replace Gmail's SMTP server with the dedicated Vijjana email server. The EmailTemplates class generates email templates such as Registration Email Template, Keyword Subscription Email Template etc.

# Chapter 6 Implementation

---

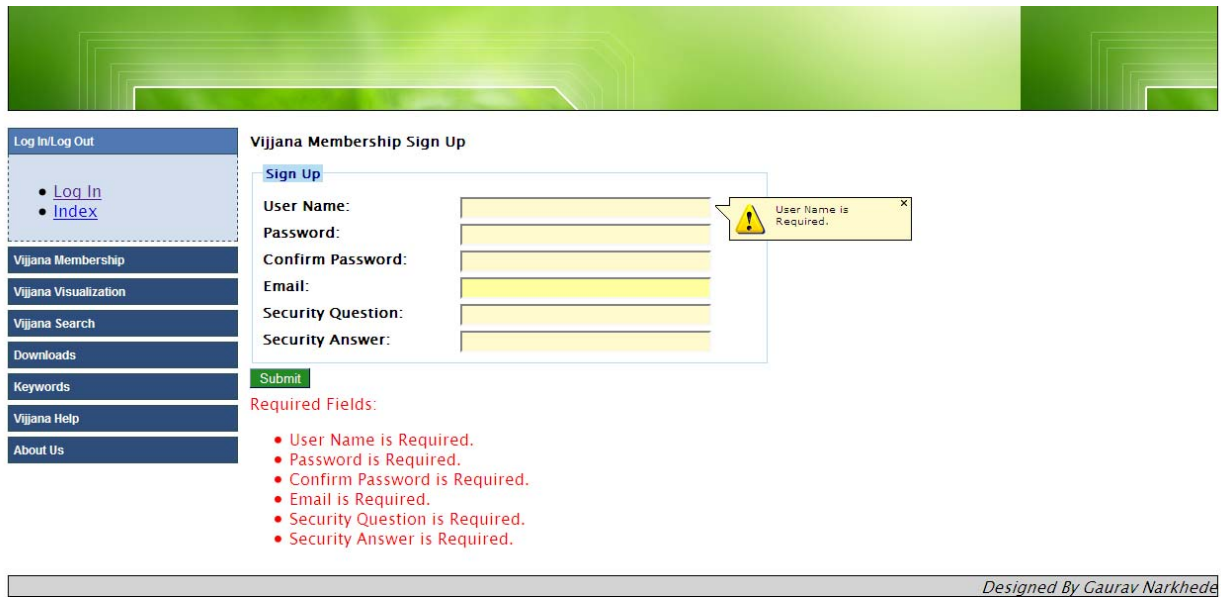
The implementation details are as follows. For each functionality, the corresponding files are listed and their implementation details are discussed. The Forms are always presented as \*.aspx files and the implementation code, validation, Database Connectivity etc are presented as c# code in the corresponding \*.aspx.vb file.

## 6.1 New User Registration



The screenshot shows a web page titled "Vijjana Membership Sign Up". On the left side, there is a navigation menu with the following items: "Log In/Log Out", "Log In", "Index", "Vijjana Membership", "Vijjana Visualization", "Vijjana Search", "Downloads", "Keywords", "Vijjana Help", and "About Us". The main content area contains the sign-up form. At the top of the form is a "Sign Up" link. Below it are six input fields: "User Name:", "Password:", "Confirm Password:", "Email:", "Security Question:", and "Security Answer:". The "Email:" field is highlighted in yellow. At the bottom of the form is a green "Submit" button.

Figure 5: Vijjana Membership Sign Up



*Figure 6: Vijjana Membership Sign Up with Validation*

The following files are used for New User account creation functionality.

Membership.aspx

Membership.aspx.vb

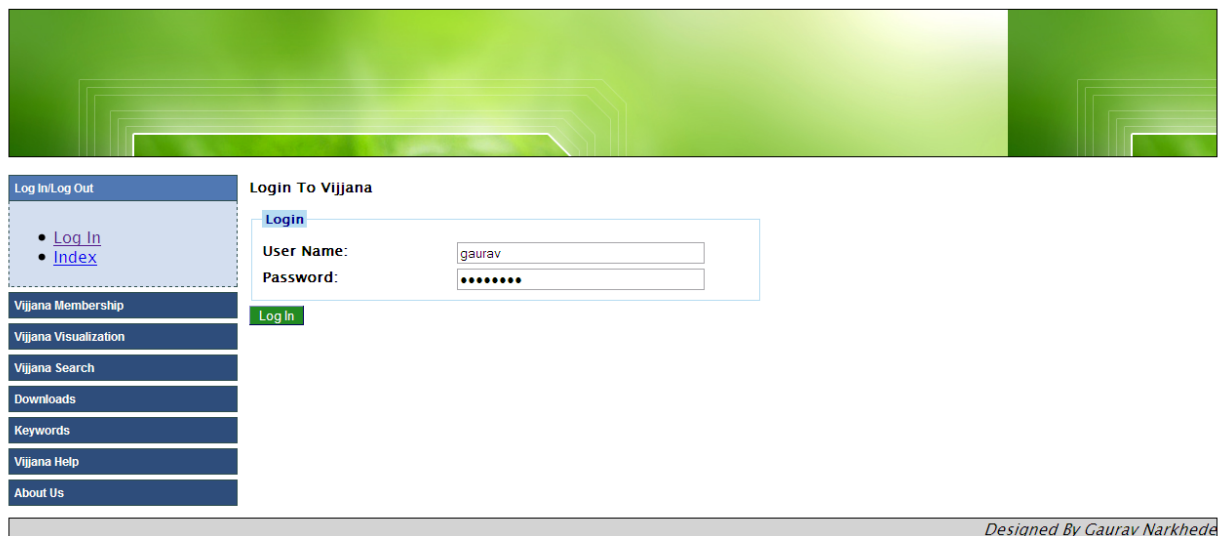
The New User is presented with a form (Membership.aspx) to add the following details: User Name, Password, Confirm Password, Email, Security Question, and Security Answer.

The values are passed on to Membership.vb file once the submit button is clicked. The Membership.vb does the following validations:

- ✓ All fields are mandatory and no field can be empty
- ✓ User Name can only contain alphanumeric characters. Valid characters are uppercase or lowercase alphabets and numbers.

- ✓ If a User Name already exists, it will display the error message "User Name already exists."
- ✓ If Email already exists, it will display the error message "Email already exists".
- ✓ Email should be of format: [alphanumeric@sample.com](mailto:alphanumeric@sample.com)
- ✓ If Password is empty an error message "Password is required" will be displayed
- ✓ After validations if the entered data is good, then the message "User account successfully created" is displayed. Upon successful account creation an Email is sent to the user on specified address with all the account details.

## 6.2 Login



*Figure 7: Vijjana Login Page*

The following files are used for Login function.

Login.aspx

Login.aspx.vb

The Login.aspx file is the login page for both the administrator and the students. The person would need to enter his login and password in this page. These are passed on to the Login.aspx.vb file. The Login.aspx.vb file is used to verify the login, password and status of the person. The login name, password and status are maintained in a MySQL Database Table "tblMembership".

The Login.aspx.vb file verifies that the combination of User Name and Password exists in the database table. If the user logs in as an administrator then that user is assigned an 'Admin' role, and is taken to the page "AdminPanel.aspx", which is the main page for the administrator. If the user logs in as a regular user then that user is assigned with 'Regular' role, and the user is taken to the page "Index.aspx", which is the main page for the Vijjana Website. If the user name and password do not match, then the error message "Incorrect User Name or Password" is displayed. The user then has to either check his login for spelling mistakes or create a New User account by clicking on the New User Registration link from the left hand side main menu.. This link redirects to "Membership.aspx" form.

## 6.3 Edit Account Information

Log In/Log Out

- [Log In](#)
- [Index](#)

Vijjana Membership

Vijjana Visualization

Vijjana Search

Downloads

Keywords

Vijjana Help

About Us

Vijjana Edit Membership Information

[Edit Membership](#)

User Name: gaurav

Password:

Confirm Password:

Email: gnarkhed@mix.wvu.edu

Security Question: What?

Security Answer: Nothing

Update

Designed By Gaurav Narkhede

*Figure 8: Vijjana Edit Account Information Page*

The following files are used for Edit Account Information Functionality.

EditMembership.aspx

EditMembership.aspx.vb

The New User is presented with a form (EditMembership.aspx) to add the following details: User Name (Disabled), Password, Confirm Password, Email, Security Question, and Security Answer.

The values are passed on to EditMembership.vb file once the submit button is clicked. The EditMembership.vb does the following validations:



- ✓ All fields are mandatory and no field can be empty
- ✓ User Name can only contain alphanumeric characters. Valid characters are uppercase or lowercase alphabets and numbers.
- ✓ If a User Name already exists, it will display the error message "User Name already exists."
- ✓ If Email already exists, it will display the error message "Email already exists".
- ✓ Email should be of format: [alphanumeric@sample.com](mailto:alphanumeric@sample.com)
- ✓ If Password is empty an error message "Password is required" will be displayed
- ✓ After validations if the entered data is good, then the message "User account information successfully updated" is displayed. Upon successful account information updating an Email is sent to the user on specified address with all the new account details.

## 6.4 Password Recovery

Log In/Log Out

- [Log In](#)
- [Index](#)

Vijana Membership

Vijana Visualization

Vijana Search

Downloads

Keywords

Vijana Help

About Us

Vijana Password Recovery

Password Recovery

User Name:

Security Question:

Security Answer:

Next Recover Password

Designed By Gaurav Narkhede

### *Vijana 9: Vijana Password Recovery Page*

The following files are used to implement Password Recovery functionality.

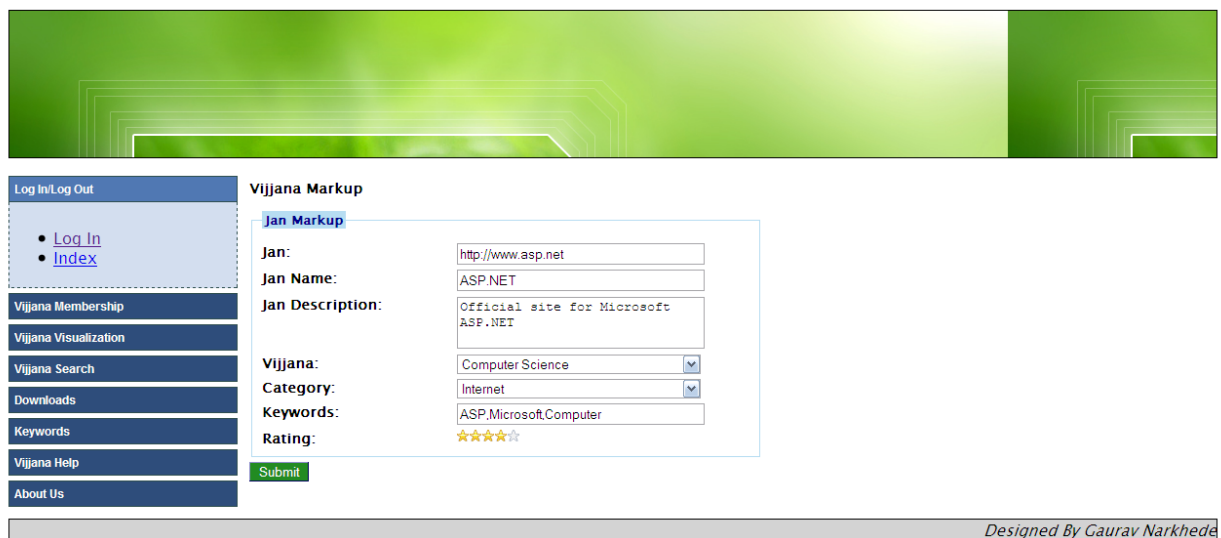
PasswordRecovery.aspx

PasswordRecovery.aspx.vb

The user is presented with the PasswordRecovery.aspx form and asked to enter the User Name. If the User Name is incorrect then "Incorrect User Name" message is displayed. If the User Name is correct then User is presented with the Security Question entered during Account Registration Process and asked to enter the answer for the question. If the answer is incorrect the "Incorrect answer" message is displayed and user is asked again to enter the correct answer. If the answer is correct then the user's

password sent to the email address entered by the user during registration process and user is informed that his/her password has been sent to their email address with the help of message box.

## 6.5 Markup



The screenshot displays a web application interface for adding a new markup entry. The page features a green header with a circuit-like pattern. On the left, there is a navigation menu with links for 'Log In/Log Out', 'Log In', 'Index', 'Vijjana Membership', 'Vijjana Visualization', 'Vijjana Search', 'Downloads', 'Keywords', 'Vijjana Help', and 'About Us'. The main content area is titled 'Vijjana Markup' and contains a form with the following fields: 'Jan' (text input with 'http://www.asp.net'), 'Jan Name' (text input with 'ASP.NET'), 'Jan Description' (text area with 'Official site for Microsoft ASP.NET'), 'Vijjana' (dropdown menu with 'Computer Science'), 'Category' (dropdown menu with 'Internet'), 'Keywords' (text input with 'ASP,Microsoft,Computer'), and 'Rating' (radio buttons for 1 to 5 stars). A green 'Submit' button is located at the bottom of the form. The footer of the page reads 'Designed By Gaurav Narkhede'.

*Figure 10: Vijjana Markup Page*

The following files are used to implement Jan Markup functionality.

Markup.aspx

Markup.aspx.vb

The user is presented with Markup.aspx form to add the following details Jan (URI), Jan Name, Jan Description, Vijjana, Category, Keywords, and Rating.

The Markup.aspx form is actually invoked by the user by clicking on the

Markup button in the browser which comes with the Vijjana Browser Extensions. Jan (URI) and Jan Name fields are automatically filled for the user.

The values are passed on to Markup.vb file once the submit button is clicked. The Markup.vb file does following validations:

- ✓ All fields are mandatory and no field can be empty.
- ✓ User is logged in.
- ✓ The Jan URI in the Jan field presents a valid URI format.
- ✓ All the keywords are separated by comma.

## 6.6 Search

The screenshot displays the Vijjana Search Page. At the top, there is a search bar with a 'Search' button. Below the search bar, the page is divided into three main sections: a navigation menu on the left, a search results area in the center, and a footer at the bottom.

**Navigation Menu (Left):**

- Log In/Log Out
  - Log In
  - Index
- Vijjana Membership
- Vijjana Visualization
- Vijjana Search
- Downloads
- Keywords
- Vijjana Help
- About Us

**Search Results (Center):**

**Google Search**

Google search is a Web search engine owned by Google, Inc., and it is the most used search engine on the Web. Google receives several hundred million queries each day through its various services. The domain google.com attracted at least 135 million U.S. visitors in May 2008.

Submitted By	Date Created	Markup Count	Average Rating
<a href="#">gaurav</a>	8/20/2008 3:46:47 AM	1	5
<a href="#">Show Comments</a>	<a href="#">Save This</a>	★☆☆☆☆	

**Yahoo Search**

A Web directory created by a couple of guys from Stanford who now have more money than the entire state of Arkansas. Rumor has it they own one business suit between them. Their site is constantly updated and provides an easy way of finding almost any Web page.

Submitted By	Date Created	Markup Count	Average Rating
<a href="#">gaurav</a>	8/20/2008 3:47:58 AM	1	4
<a href="#">Show Comments</a>	<a href="#">Save This</a>	★☆☆☆☆	

**AOL Search**

America Online. An Internet provider and one of the most popular on-line services, offering, for example, E-mail, news wire reports, and user forumsStanford who now have more money than the entire state of Arkansas. Rumor has it they own one business suit between them. Their site is constantly updated and provides an easy way of finding almost any Web page.

Submitted By	Date Created	Markup Count	Average Rating
<a href="#">gaurav</a>	8/20/2008 3:48:59 AM	1	3
<a href="#">Show Comments</a>	<a href="#">Save This</a>	★☆☆☆☆	

Navigation buttons: [First](#) [Previous](#) [Next](#) [Last](#) 1

**Footer (Bottom):** Designed By Gaurav Narkhede

Figure 11: Vijjana Search Page

The following files are used to implement Search functionality.

Search.aspx

Search.aspx.vb

The user is presented with Search.aspx form which mainly consists of three regions as above:

### **Tree View**

Tree View shows all the Vijjanas and their categories to the user in hierarchical way. Whenever a user clicks on the category, corresponding Jans are displayed on the right hand side region of the Tree View.

### **Search Results**

Search Result consists of Jans. For each Jan in the search result following information is displayed: Jan Name, Jan URI, Jan Description, User Name, Date Created, Markup Count, and Average Rating. Along with this information user can also Show/Hide comments for particular Jan, save Jan to his personal network and rate Jan using five star rating system.

### **Search Box**

Search box located at the top of Search.aspx form is an alternative to Tree View. User can type a keyword or a phrase into the search box and press search button. This will display corresponding Jans into the Search Results area.

## 6.7 Add Keywords, Rename Keywords or Remove Keywords

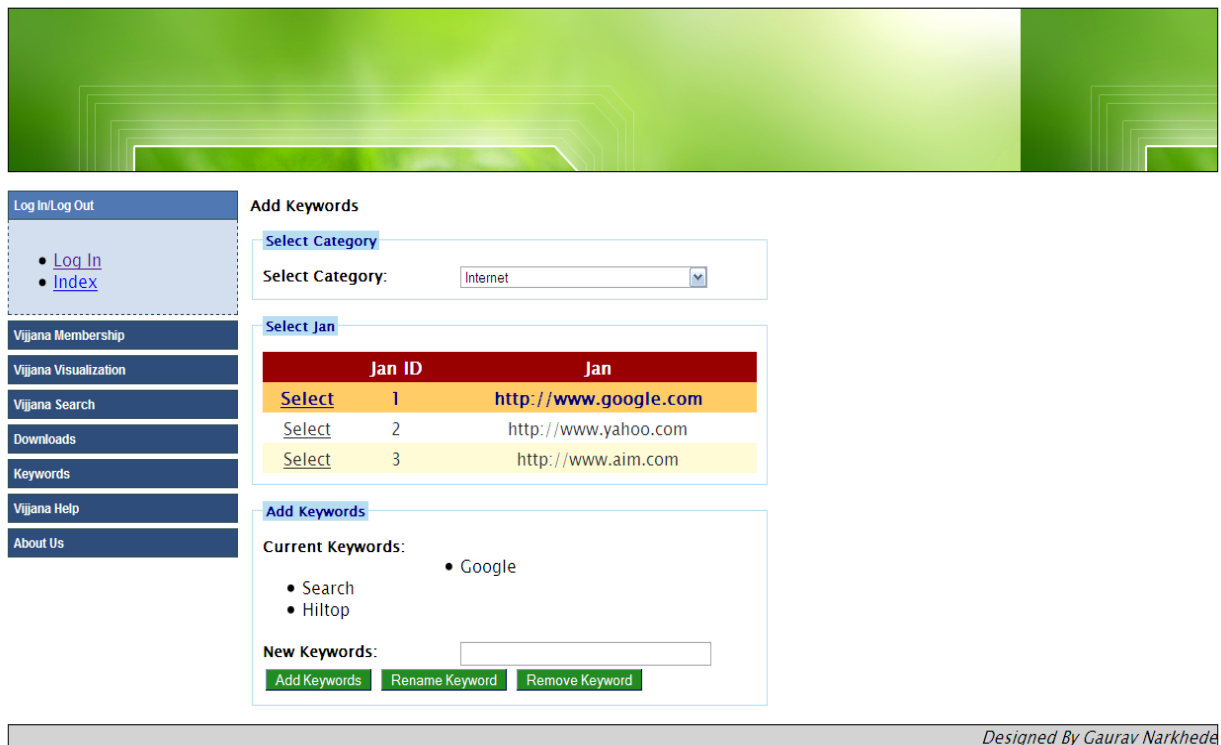


Figure 12: Vijjana Keyword Manager Page

The following files are used to implement Keyword Management Functionality.

KeywordManager.aspx

KeywordManager.aspx.vb

The user is presented with KeywordManager.aspx form. The user is asked to select the category from the category drop down list. As soon as user selects the category, a grid view is shown to the user containing all the Jans in the selected category. Then user is asked to select a Jan from the grid view to

which user wants to add keywords. Once the user selects a Jan, corresponding keywords are displayed as a selectable list. The user can select any keyword from this list and Rename or Remove it. The user can also add new keywords by typing keywords in the New Keyword Textbox.

## 6.8 Keyword Subscription

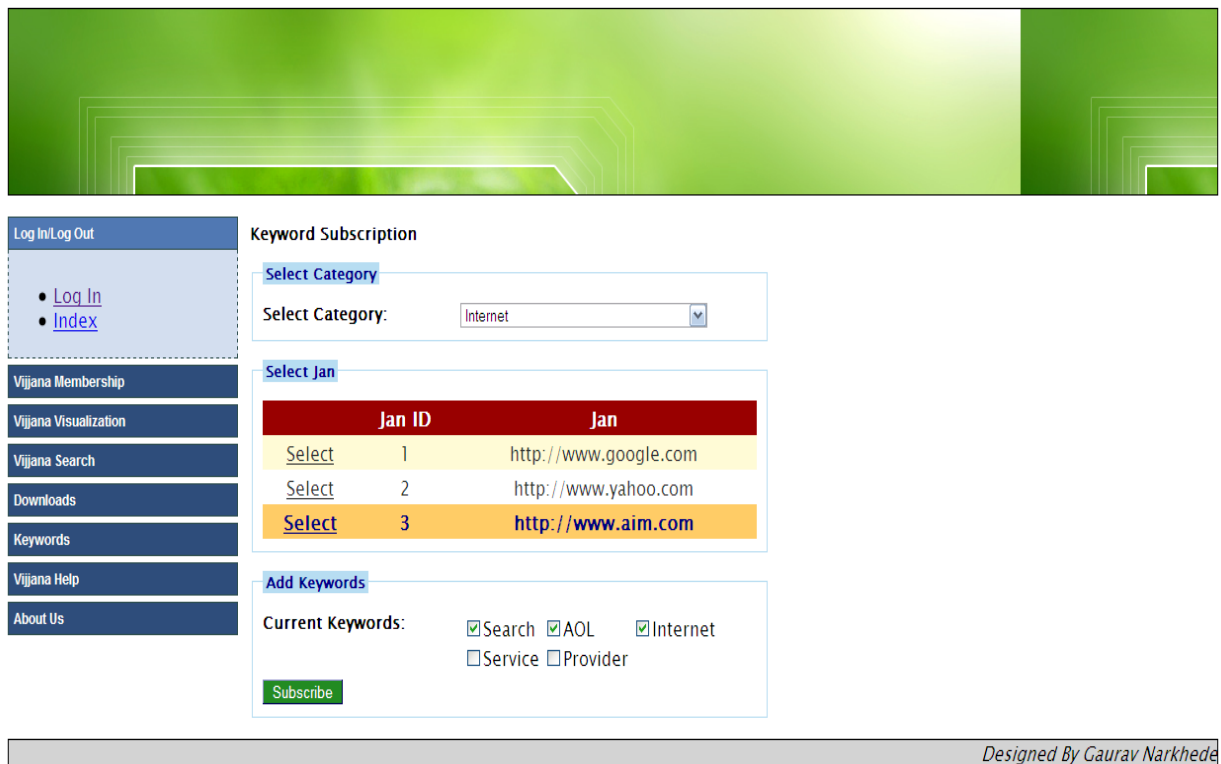


Figure 13: Vijjana Keyword Subscription Page

The following files are used to implement Keyword Subscription Functionality.

KeywordSubscription.aspx



## KeywordSubscription.aspx.vb

The user is presented with KeywordManager.aspx form. The user is asked to select the category from the category drop down list. As soon as user selects the category, a grid view is shown to the user containing all the Jans in the selected category. Then user is asked to select a Jan from the grid view. Once the user selects a Jan, corresponding keywords are displayed as a check box list. The user can select as many keywords as he/she wants from this list and press Subscribe button.

# Chapter 7 Future Work and Conclusion

---

## 7.1 Future Work

- ✓ The current implementation of Organizing Agent can be extended to include functionality for integrating Vijjana services with Microsoft Office Products and Google Documents.
- ✓ The Organizing Agent can be extended to use the data exposed by Google Search API, Yahoo Search API, Digg.com, del.icio.us and many other web sites.
- ✓ The process of classifying and organizing the Jan can be refined.
- ✓ The UI can be made more users friendly.

## 7.2 Conclusion

In this paper we saw the implementation of an Organizing Agent for Vijjana Framework. The Organizing is responsible for organizing a Jan into the evolving Vijjana, recording information about Jan in a Provenance File, automatic notifications and responding to the events generated by the Visualization Agent. The Organizing Agent acts as a middleware application between Discovery Agent and Visualization Agent. We saw how Organizing Agent provides convenient way of managing the Vijjana Database, and

performing CRUD as well as advance operations on database. We also saw how Organizing Agent can serve as basic framework for other Vijjana agents such as Search Agent, Visualization Agent etc. The Organizing Agent is implemented as separate class library project in VB.NET. Therefore along with Website it can also be used to develop dedicated client application for Vijjana.

# References

---

1. Vijjana: A Pragmatic Model for Collaborative, Self-organizing, Domain Centric . Reddy, Dr. Ramana. Morgantown : IKE08, 2008.
2. Creating Data Access Layer: Scott Mitchell  
<http://msdn.microsoft.com/en-us/library/aa581778.aspx>
3. Creating Business Logic Layer: Scott Mitchell  
<http://msdn.microsoft.com/en-us/library/aa581779.aspx>
4. Ivan Herman, (W3C) Semantic Web Activity Lead. W3C Semantic Web. W3C. [Online] 1994-2008. <http://www.w3.org/2001/sw/>.
5. What is a Semantic Web?  
<http://www.w3schools.com/semweb/default.asp>
6. How search engine works?  
<http://www.lib.berkeley.edu/TeachingLib/Guides/Internet/SearchEngines.html>
7. Ontology and Agent Based Searching  
[http://www.iaeng.org/IJCS/issues\\_v33/issue\\_1/IJCS\\_33\\_1\\_22.pdf](http://www.iaeng.org/IJCS/issues_v33/issue_1/IJCS_33_1_22.pdf)
8. Professional ASP.NET by Bill Evjen, Scott Hanselman, Farhan Muhammad, S. Srinivasa Sivakumar, Devin Rader
9. Foundations of ASP.NET AJAX: Build AJAX applications in ASP.NET by Robin Pars, Laurence Moroney and John Grieb
10. Data Access Layer: Damon Armstrong  
<http://www.simple-talk.com/dotnet/.net-framework/.net-application-architecture-the-data-access-layer/>

# Appendix A: Presentation Slides

---

## ORGANIZING AGENT FOR VIJJANA

**Chair**

**Dr. Yenumula. V. Reddy,  
Ph.D.**

**Committee Members**

**Dr. James D. Mooney, Ph.D.  
Dr. Sumitra Reddy, Ph.D.**

**Presented By**

**Gaurav Narkhede**

Lane Department of Computer Science and Electrical Engineering  
WEST VIRGINIA UNIVERSITY

### OUTLINE

- Organization
- Design of Organizing Agent
- Implementation of Organizing Agent
- Future work and conclusion
- References

## PART-1 ORGANIZATION

- What is Organizing Agent?
- Organizing Agent in Vijjana Workflow
- Functions of Organizing Agent

## WHAT IS ORGANIZING AGENT?

- The Organizing Agent is a component in the Vijjana which is responsible for organizing a Jan into the evolving Vijjana, recording information about Jan in a Provenance File, automatic notifications and responding to the events generated by the Visualization Agent.
- The Organizing Agent acts as a middleware application between Discovery Agent and Visualization Agent.

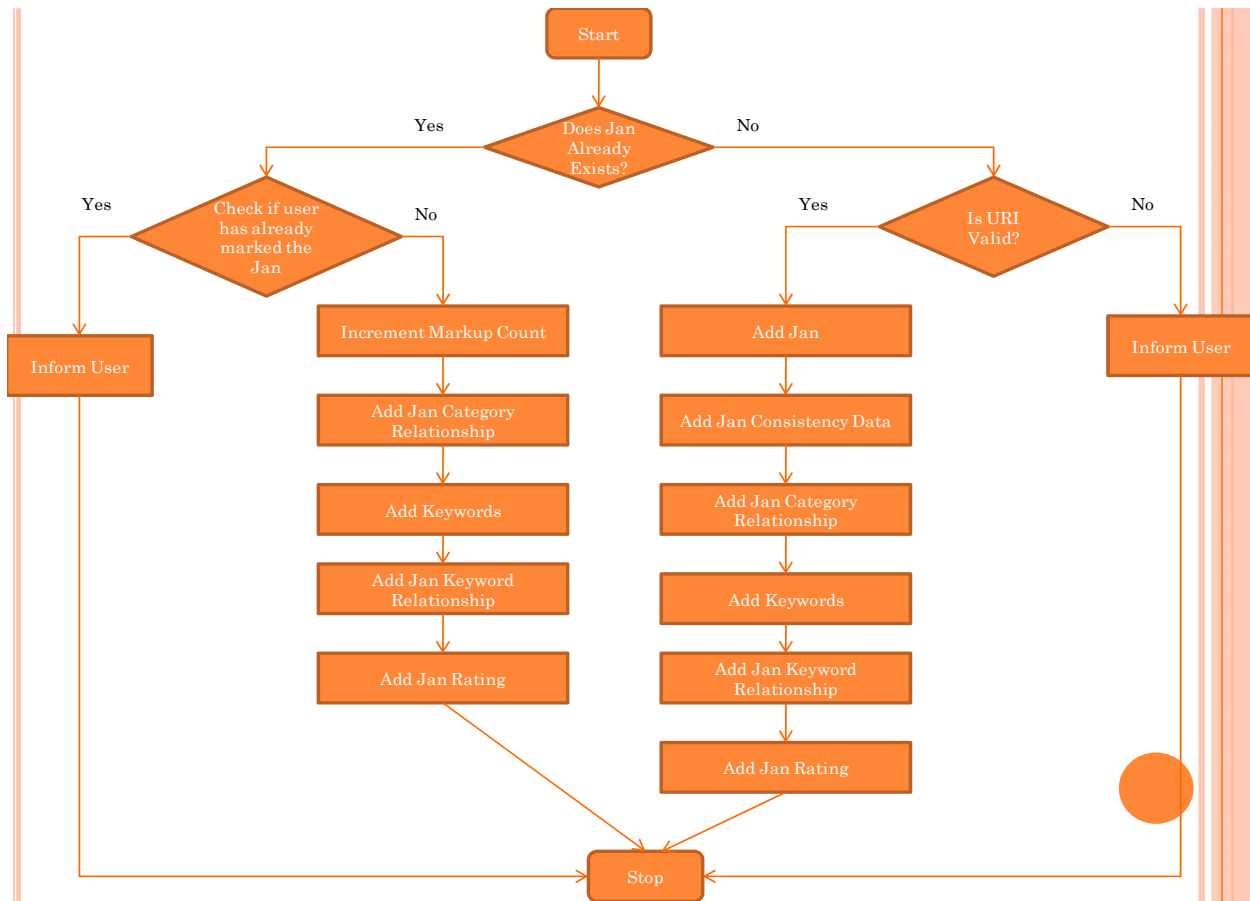
## ORGANIZING AGENT IN VIJJANA WORKFLOW

Organizing Agent comes into action during following scenarios-

- During Jan markup process
- During Visualization
- While entering consistency records
- During administration process

## FUNCTION OF ORGANIZING AGENT

- Organization of Jan
- Maintaining provenance file
- Automatic notifications about incorporation of new Jan
- Respond to the events generated by Visualization Agent



## PART-2 DESIGN OF ORGANIZING AGENT

- What is a Software Layer?
- Vijjana is designed to have layered Software Architecture. It has four main layers Data Layer, Data Access Layer, Business Logic Layer, and Presentation Layer.
- Data Access Layer and Business Logic Layer together provide the functionality of Organizing Agent.



## ADVANTAGES OF LAYERED ARCHITECTURE

- Availability and Scalability
- Data abstraction
- Easy maintenance
- Reduces coupling between User Interface, Business Logic and Database.
- allows software application development to be more effectively split into teams

## DATA ACCESS LAYER

- The main objective of Data Access Layer is to provide data from database to Business Objects without using the database specific code such as SQL Queries, ADO.NET code etc.
- Data Access Layer perform some of the very common tasks on the database on behalf of their caller –
  - Create new records in the database
  - Retrieve records from database based on some criteria
  - Update records
  - Delete records

## BUSINESS LOGIC LAYER

- The Business Logic Layer encapsulates all the business logic and enforces data validation and business rules.
- For our Vijjana Web Application we want the User Name field of Membership table to 20 characters long and it should only contain Uppercase or Lowercase alphabets. This example comes under validation rules.
- We might want to disallow unregistered users from accessing certain data. This comes under business rules.

## PART-3 IMPLEMENTATION OF ORGANIZING AGENT

- Each layer is implemented as a separate namespace contained by VIJJANA namespace.
- There are nine namespaces-
  - DatabaseManipulation
  - Entities
  - DataAccessLayer
  - BusinessLogicLayer
  - ConsistencyAgent
  - Notifications
  - Enumerations
  - SearchServices
  - Email

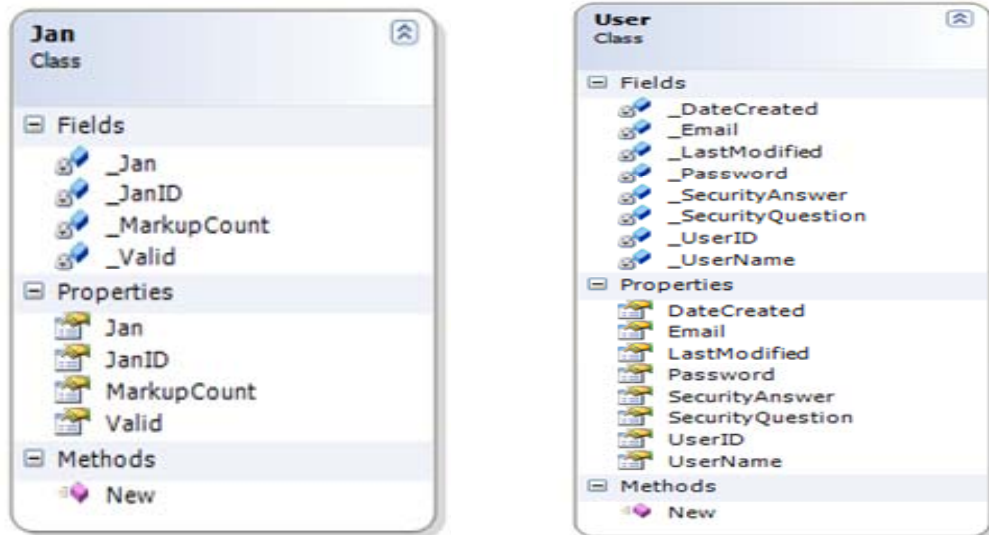
## DATABASEMANIPULATION NAMESPACE

- This namespace contains only two classes named AdapterFactory and MySQLDatabaseAdapter.
- AdapterFactory class implements the Abstract Factory Design Pattern.
- MySQLDatabaseAdapter is class that encapsulates all the code required to interact with the MySQL Database Server

## ENTITIES NAMESPACE

- The Entities namespace contains all the Business Objects such as Vijjana, User, Keyword, Jan etc.
- These Business Objects represent corresponding tables in the database.
- These objects also expose several properties corresponding to fields in the database table.
- These business objects have several overloaded constructors which are suitable for different programming situations.

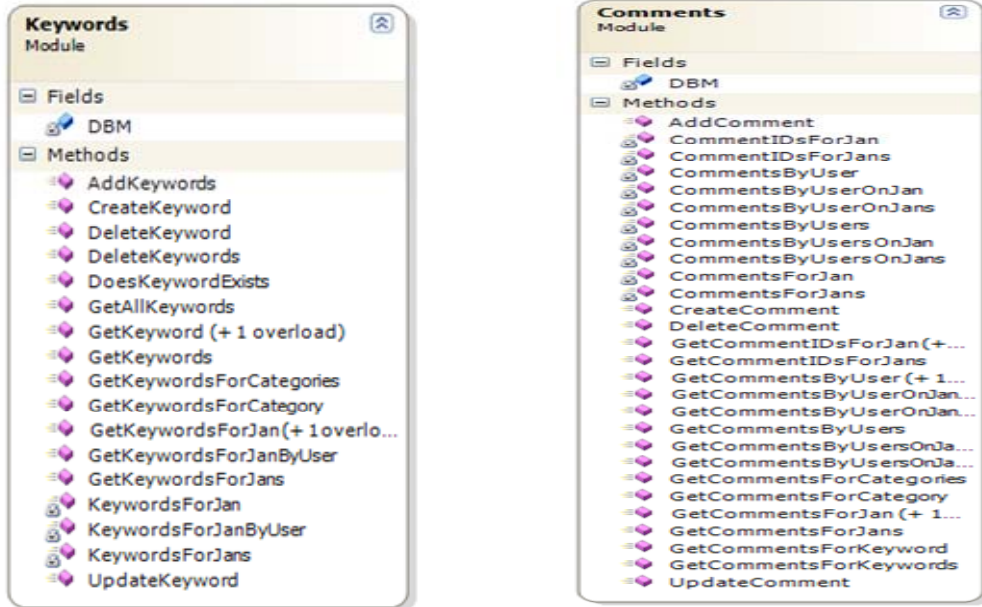
## EXAMPLE OF BUSINESS OBJECTS



## DATAACCESSLAYER NAMESPACE

- The DataAccessLayer namespace contains all the classes which provide functionality of Data Access Layer.
- Classes in DataAccessLayer namespace return instances of Business Objects as result which removes dependence of a Web Application on database specific data types.
- Each Business Object in the result set represents a row in database table. Any operation performed on the object reflects in the corresponding database table row.

## EXAMPLE OF DAL OBJECT



## BUSINESSLOGICLAYER NAMESPACE

- The BusinessLogicLayer namespace contains all the classes which provide functionality of Business Logic Layer.
- Methods in these classes accept data from presentation layer, validate this data against the defined validation rules, apply business rules and insert data into the database. Same sequences of operations are performed in the reverse order while retrieving the data from the database.
- also takes care of Exception Handling, transactions and rollbacks.

## CONSISTENCYAGENT NAMESPACE

- The ConsistencyAgent namespace contains class named URIVValidation that validates URI of the Jan and also periodically checks the HTTP status of the URI.
- The URIVValidation class also checks the HTTP status of the URI to check its existence. An URI is considered to be in existence only if it has an HTTP status code equal to 200 (OK).
- The count of number of times an URI found nonexistent is kept into the database. If this count increases to a certain value say five the URI and its related information is automatically deleted from the database.

## EMAIL NAMESPACE

- It contains two classes namely VijjanaEmail and EmailTemplates.
- VijjanaEmail class performs the task of sending emails to users. This class uses Gmail's SMTP server, port 587 and 'Admin.Vijjana@gmail.com' as username. In the future we can replace Gmail's SMTP server with the dedicated Vijjana email server.
- The EmailTemplates class generates email templates such as Registration Email Template, Keyword Subscription Email Template etc.

## PART-4 FUTURE WORK AND CONCLUSION

### Future Work

- Integration with Microsoft Office
- Use of more structured Knowledge Bases such Google, Yahoo, Digg.com, del.icio.us, Connotea etc

### Conclusion

- We saw how Organizing Agent provides convenient way of managing the Vijjana Database, and performing CRUD as well as advance operations on database. We also saw how Organizing Agent can serve as basic framework for other Vijjana agents such as Search Agent, Visualization Agent etc.

## REFERENCES

- Vijjana: A Pragmatic Model for Collaborative, Self-organizing, Domain Centric . Reddy, Dr. Ramana. Morgantown : IKE08, 2008.
- Creating Data Access Layer: Scott Mitchell  
<http://msdn.microsoft.com/en-us/library/aa581778.aspx>
- Creating Business Logic Layer: Scott Mitchell  
<http://msdn.microsoft.com/en-us/library/aa581779.aspx>
- Professional ASP.NET by Bill Evjen, Scott Hanselman, Farhan Muhammad, S. Srinivasa Sivakumar, Devin Rader
- Data Access Layer: Damon Armstrong  
<http://www.simple-talk.com/dotnet/.net-framework/.net-application-architecture-the-data-access-layer/>