Anuvadaksh : A Real Integration

ANUVADAKSH : A REAL INTEGRATION

AAI Group, C-DAC Pune, EILMT Consortia

With national level objective, DeitY has put a step forward for collaborating all research across the India in the direction of machine translation and language technology. Resulting formation of consortium with various academic research institutes using their best expertise in the domain. We, the consortia of 13 institutes under TDIL, are part of this mission for "EILMT -English to Indian languages machine ranslation system": pertaining to English to Hindi, Marathi, Urdu, Bangla, Oria, Tamil, Gujrathi and Bodo Language, covering the selected domains of Tourism & Health care. This is a Multi-Lingual, Multi-platform & Multi engine hybrid System targeted to serve the nation by breaking the barrier of languages through its services.

Technology Development for Indian Languages (TDIL) Program initiated by the Department of Electronics and Information Technology (DeitY), Ministry of communications & Information Technology, Government of India has the objective to make common people available with machine interface in one's own language.

At a early glance, the task seemed little difficult to integrate the long year of experiments and researches of all academic institutes into a single system. There was a effort to have minimal changes into basic operating of the system(Fig.1Architecture Layer).

The cooperation from consortia played a great team work to collate all the research components and put things as a compiled model. This approach have 4 layers at architecture level: Communication Layer, Application Layer and Business Logic Layer.

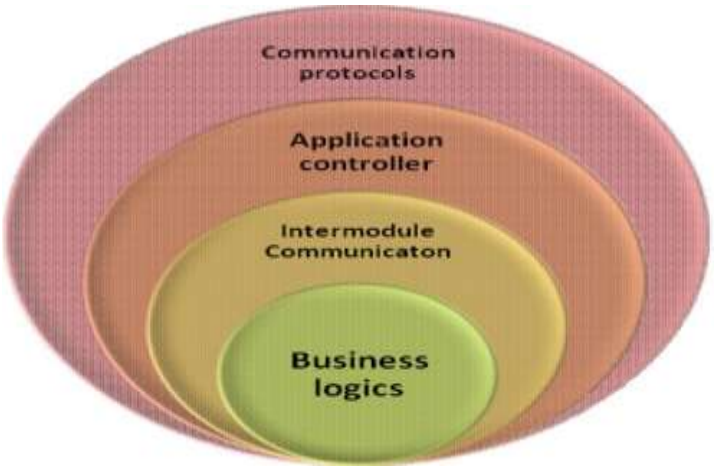

Fig. - 1 Architecture Layer

1. COMMUNICATION LAYER

1.1 NETWORK ARCHITECTURE- Anuvadaksh acts as a peripheral on machine translation platforms. This system is used as a Plug and Play in various broad level Applications. In this thin-client/thick-server design, users of this system (clients) shall use a standard browser to access the translation services of server. Clients submit the Text/Documents to the server, the job of translation is carried out at the server and the server renders translated text/documents to the clients. The EILMT application software and database consisting of Grammar Database and Lexicon Database is residing on EILMT server.

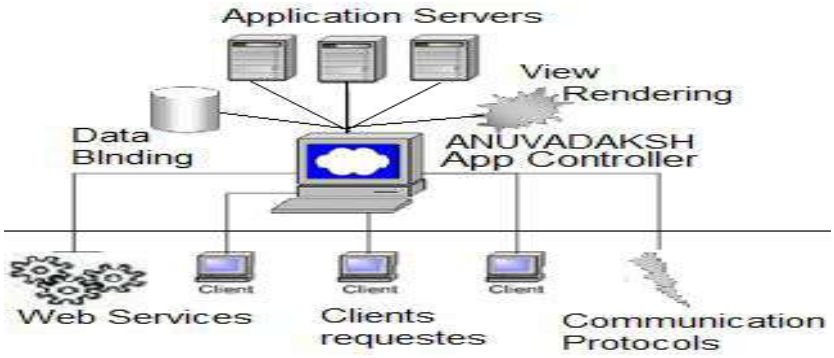

Fig. 2 - Network Architecture

Centralized design of EILMT is ideal for Internet clients who submit their English documents to a powerful EILMT server where the job of translation is performed and translated Indian Language document is sent back to the clients. Therefore, even low-end system with Internet connectivity can also avail the facility to translate the documents from remote place. This seems as an optimal solution for sharing translation-system resources.

1.2 COMMUNICATION INTERFACES TDIL- Unified User Interface Act as a leading and frontline participator and contributor for the implementation of Unified User Interface for three consortia machine Translation Projects under Technology Development for Indian Languages (TDIL) DIT. Here, Anuvadaksh plays a vital role as an application resource. We have integrated Anuvadaksh with eight Indian languages to get the translations in this common unified interface.

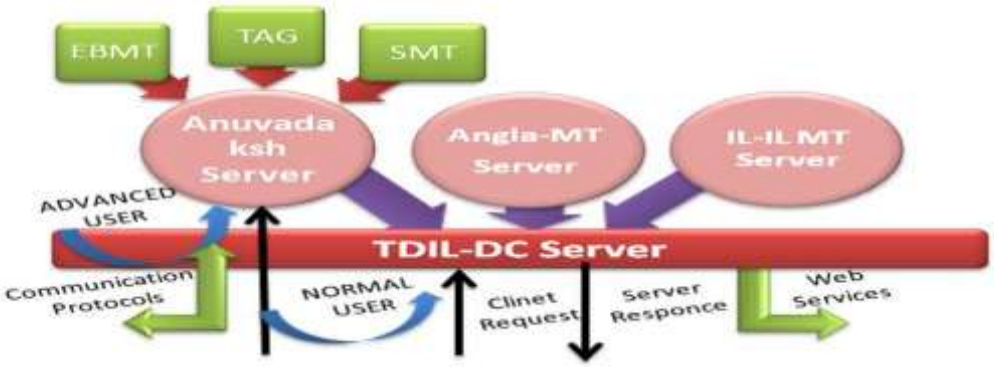

Fig. -3 Communication Interface

Communication protocols are maintained for interaction with TDIL Common GUI. TDIL-DC is portal where all consortia based engines are placed together as a consolidated Multilingual Machine Translation System. An http connection protocol are established to have an interaction between the client and the server.

1.3 HTTP PROTOCOL:

Hypertext Transfer Protocol (HTTP) is the communication protocol used to exchange information between a client system and a Web server across a TCP/IP connection. We have referred this interchange in the system. Anuvadaksh is an application-level protocol used by the client and server. We have staggered the requests instead of transmitting all at the same instant. Anuvadaksh is ported on real server by same means.

1.4 WEB SERVICES:

In order to make Anuvadaksh ubiquitous we have provided it in the form of Web Service. Web Services are designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine process able format. Due to this now Anuvadaksh can be used at all major and different platforms and also as an open protocol. Using system as a Web service presents with an opportunity to take advantage of services offered by others and the opportunity to make our application available to others as a Web service very easily. Anuvadaksh as a web service have given benefits in exposing the existing functions on to network. It has increased the interoperability and it is cognized as a Standardized Protocol at every platform.

2. APPLICATION LAYER

System is implemented with J2EE environment with JBOSS Application Server. J Boss Application Server is the open source implementation of the Java EE suite of services. It is easy-to-use server architecture and high flexibility with a customizable middleware platform. System is on struts framework. Apache Struts is an open-source web application framework for developing Java EE web applications. It uses and extends the Java Servlet API to encourage developers to adopt a model view-controller (MVC) architecture. Running TAG Parser as a web application may cause huge resource & memory requirement so system is experimented with multithreaded and multi-core environment to improve response time and through put of system. System is ported to JVM with 64 bit Operating System and J Boss server with 8 GB heap space size so the execution of complex and lengthier jobs also is simpler to run.

2.1 TRANSLATION SYSTEM ARCHITECTURE

The Workflow of English to Indian Languages Machine Translation consists of five main modules namely User Log Module, Pre-Processing Module, Parsing and Generation Module, Collation and Ranking Module and Post Processing Module. EILMT, being a web based Project, has a responsibility to work with multiple users and multiple requests simultaneously. As an application Server, J Boss 5.0.0 is chosen which is devoted to the efficient execution of procedures. Proper session handling is a big task, when the system is working in multi-user scenario, so we have used a robust Database I/O instead of memory blocking file/exe. System is working with EJB 2.0 (Enterprise Java Bean) which enables rapid and simplified development of distributed, transactional, secure and portable applications. System moved to Struts frame 1.1, it encourages consistent use of MVC throughout the application and provides utility

classes to handle many of the most common tasks in Web application development.

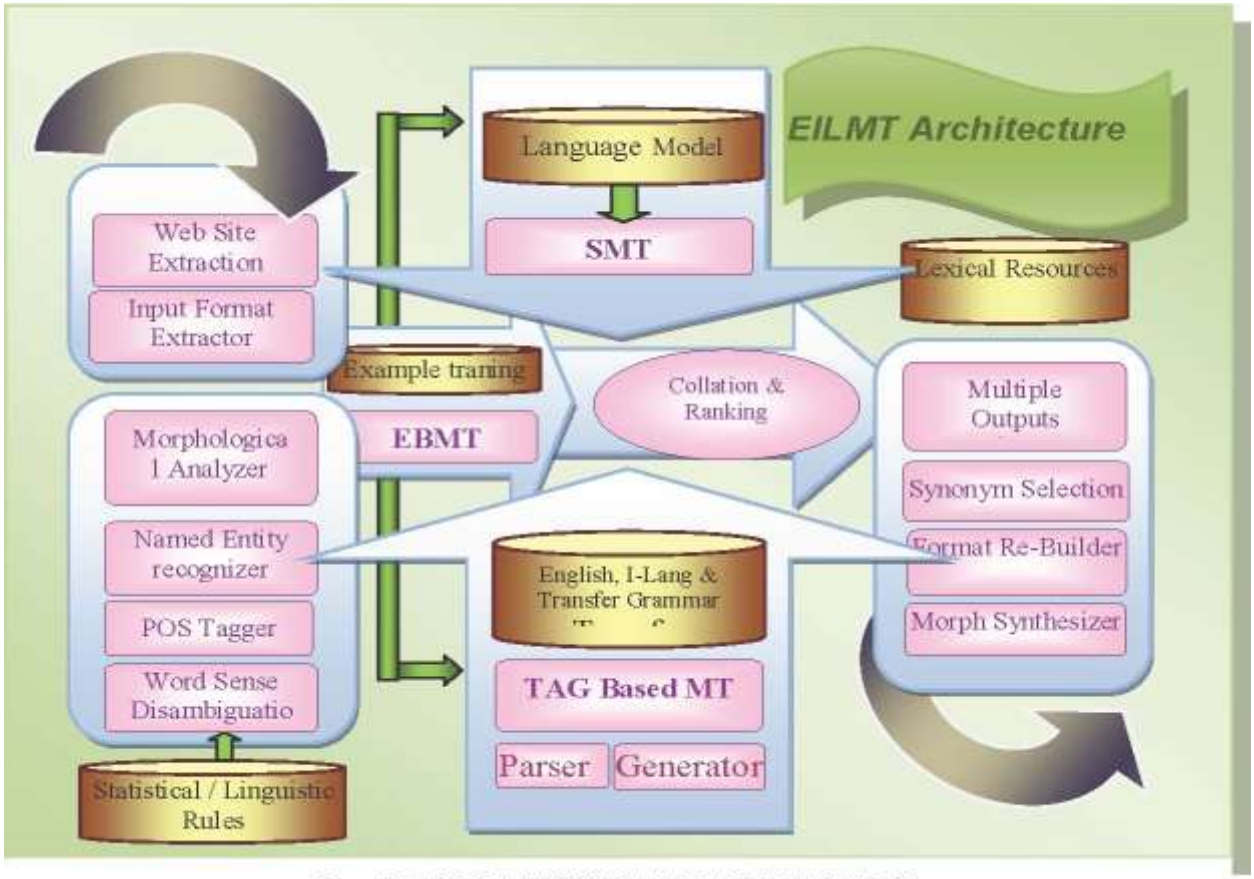

Fig. 4 TRANSLATION SYSTEM ARCHITECTURE

Centralized design of EILMT is ideal for Internet clients, who submit their English documents to a powerful multi-core EILMT Server, where the job of translation is performed. The translated document is sent back to the clients.

The System is designed to use three-translation engines working in parallel. The pre processed text is the input to the four translation engines and generated output by the all the engine goes to collation and ranking module for evaluation.

One of the translation engines of EILMT system is Example-Based Machine Translation (EBMT). This approach is characterized as it uses a bilingual corpus as its main knowledge base, at run-time. EBMT is integrated with number of knowledge resources, such as linguistics and statistics, symbolic and numerical techniques, for integration into one framework. Another translation engine being used is Statistical Translation Model (SMT), which is a mathematical model. SMT correspondences between the words in the source and the target language, which are learned from bilingual corpora on the basis of so-called alignment models.

The Tree Adjoining Grammar (TAG) is one of the four engines of the EILMT system. The TAG consists of a set of elementary trees, divided into initial and auxiliary trees. This works on tree-to-tree translation model and makes use of syntactic tree for both the source and target language. Parser and Generator modules recognize various grammatical entities in the English sentence, analyze them, represent them in different tree structures and synthesize equivalent Indian Language sentences on the basis of the derivational tree structure and the Transfer Grammar. It has two layers of multi-threaded embedding. The Server handles each client's request and maintains its session. It generates the threads to call all the engines in parallel. One of the outer layer thread calls the inner layer of multi-threading, that improves the heart of the TAG translation engine (the parser) by spawning a new thread for each sentential initial tree of a sentence, thereby building several derivational trees simultaneously.

The other outer layer threads makes a connection with SMT server and another communicates with EBMT engine. Initially EBMT engine was implemented in per with Linux platforms, which have been

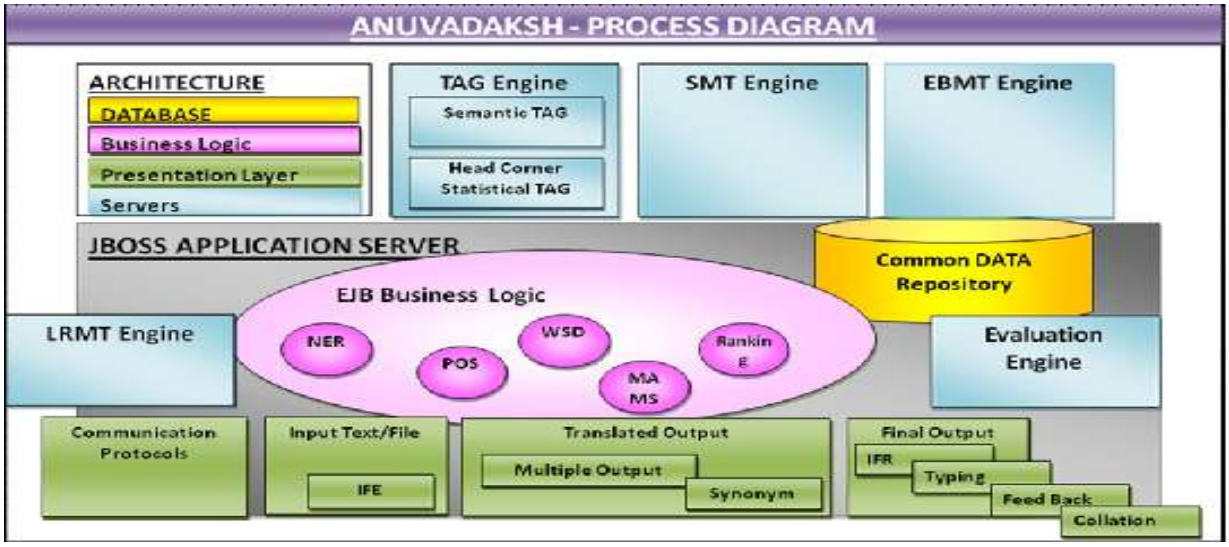

Figure 5 EILMT http communication protocols

ported to windows platform. SMT is in Java, works with Linux Operating system. There was experiment to call some engine as a web service form but due to avail more performance efficiency it is decided to keep these engines on a separate physical server having peer-to-peer connectivity. An http request made a call to these engines to perform their operations.

The Collation & Ranking Module collates and ranks the translated output generated by different translation engines associated with the system. While preparing a complete integrated system of EILMT, options such as – SOAP/RPC, calling a DLL/EXE, intermediate File I/O, http protocol request and web service has been taken into consideration. A Standard has been set to communicate through Database for intermediate I/O transfer. Post Processing Module is targeted to provide additional features for EILMT Translation engine like multiple output selection, Synonym Replacement and Format re-builder.

2.2 DATA HANDLING

As being a Multilingual Translation System, Database Data stored in the EILMT System is having eight clusters of data (Each per Language) and some common data tables required for common purpose.

The Database being used in EILMT System is MySQL 6.0. MySQL is one of the most popular databases on the Internet, which is a multithreaded, multi-user Relational database management system (RDBMS). Besides its undoubted advantages such as ease of use and relatively high performance, MySQL offers simple but very effective security mechanisms.

MySQL 6.0 supplies more reliability, performance, and ease-of-use enhancements, starting with the new Falcon transactional storage engine. MySQL 6 features ACID transaction compliant, Crash recovery, User defined table spaces, High-speed data caches, Advanced B-Tree indexes, Performance/diagnostic monitoring tables and Simplified configuration, among others. Falcon utilizes table spaces for user data storage with there being no practical limit to how many table spaces can be created and used to manage tables, indexes, and BLOB data. All table spaces feature auto-extending data files, automatic space reclamation, and compaction of data pages.

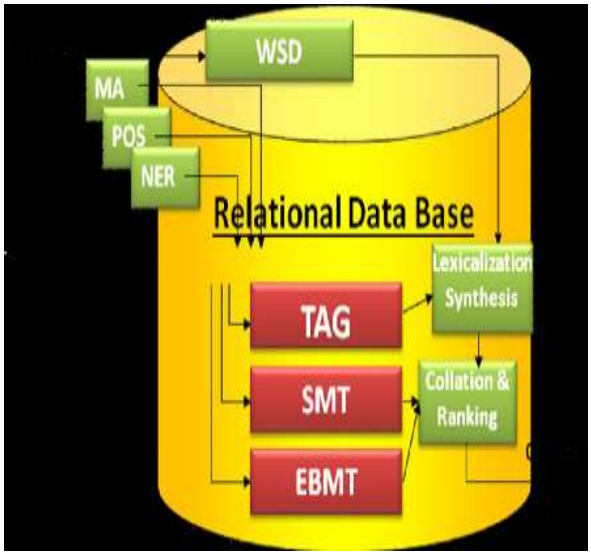

Fig. 6 Internal Intermediate Communication

3. BUSINESS LOGIC LAYER

3.1 CODE IMPLEMENTATION

The EILMT system is J2EE applications are comprised of components, containers, and services. Components are application-level components. Web components, such as Servlets and JSPs, provide dynamic responses to requests from a Web page. EJB components contain server-side business logic for enterprise applications. Web and EJB component containers host services that support Web and EJB

modules. The An Enterprise Java Bean (EJB) is a reusable, portable J2EE component, which consist of methods that encapsulate business logic. Enterprise Java Beans (EJB) is managed, server-side component architecture for modular construction of enterprise applications. Struts frame 1.1 encourages consistent use of MVC throughout the application and provides utility classes to handle many of the most common tasks in Web application development.

4. USER INTERFACES

The EILMT system provides User Log module to facilitate with user friendly Graphical User Interface (GUI). This GUI eases the registration process for new users to get registered with the system. It also provides login facility where the user is secured with his/her personal information. The user has to select the Language Pair [English-Hindi, English-Marathi, English-Bangla, English-Oriya, English-Tamil, English-Urdu, English-Guajarati & English-Bodo].

Only registered users can login into the system. After login, the GUI provides browse option through which user can upload the file for which he/she needs translation in .rtf, .xls, .txt and .html format. To start translation the user should click 'Proceed' button, which gives the translated output in the corresponding language selected by the user. This module also provides the facilities to maintain the format of the user input, which shall be used for format extraction and rebuilding in later stage. Since the communication protocols make request to EILMT Server for translation, it automatically takes care of login user and session provided by caller portal. Apart from that, a registered user can directly login to system by submitting respective user-name and password. On successful login, a translation screen appears where the facility of language and domain selection has been provided. There is a facility to enter input text in two forms. User can upload a file for translation along with entering input text in text area. On pressing the translate button, the interface makes a call to translation server and provide translated output to user sentence by sentence in his desired language. Translated outputs have various facilities for post processing like – Multiple output selection, Synonym selection, care marker and typing facility. On click over the "more" button, user may get more translations, which are the outcome of various engines. The highlighted color tokens in translated output represents synonym options, clicking on which it provides multiple meaning of corresponding tokens.

Fig. 7 Login Interface

Fig. 8 - Upload Interface

Fig. 8 - Translation Interface

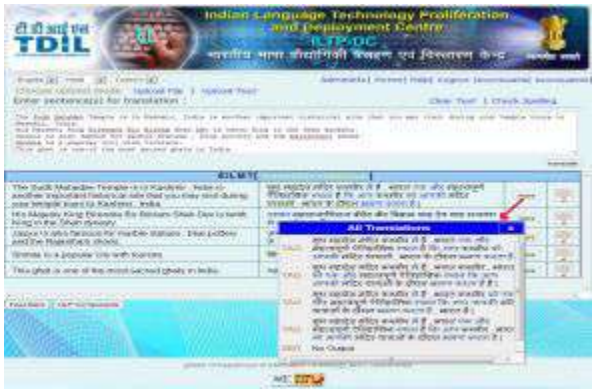

Fig. 10 Multiple output facility

Fig. 11 Synonym Selection facility

User is presented with the above screen with the final output. Help link provides information of all buttons and links existing in the system. Home link takes user to home page where user can input new sentences for translation. Download XML log downloads input and output in XML format.

Download XLS log downloads input and output in XLS format. Feedback Button allows user to provide modification or control in the process of system by its results or effects. NLP Components button provides user with different levels of output generation. Logout link automatically logs out the user from system. Our Translation System is capable of providing output of intermediates module in form of NLP components in xml standard format. It can provide tool support and facilitate consortium to access our website to download the NLP components as if when needed. Different modules which comes under NLP components are : Preprocessed Output, POS Final Output, TAG Parsed Derived Output, TAG Generated Derived Output and Translated Output. Derived and derivation Output is used to form the Tree view of the source and translated output.

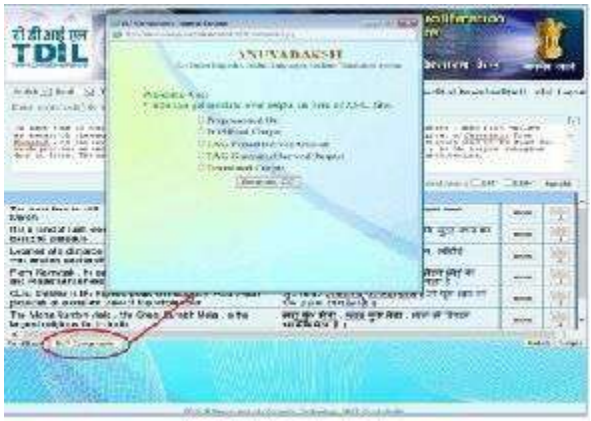

Fig. 12 NLP Components facility

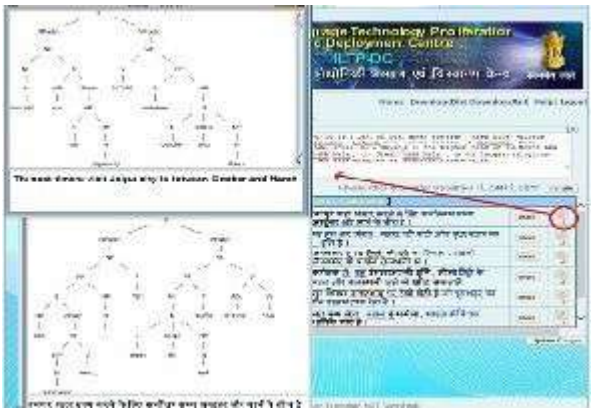

Fig. 13 Tree View facility

5. PERFORMANCE:

To speed up the online translation process we have introduced a TM, a database of derivation structure of source and target sentence, integrated at the backend of the MT system. We have implemented a fully automated TM as a part of research from a RBMT using Tree Adjoining Grammar (TAG). This TM can be reused by the same RBMT system when same set of sentences or structure is encountered. Comparing new sentence for translation to already done translations improves consistency and saves time: you can

reuse previous translations or adjust them to create new, more contextually appropriate translations. This maintains the consistency and increases the efficiency of the system.

6. HIBERNATE ON ANUVADAKSH:

Hibernate is an open source mapping tool. Hibernate maps the objects to the relational model where entities/classes are mapped to tables, instances are mapped to rows and attributes of instances are mapped to columns of table. We have implemented Hibernate framework in Anuvadaksh to increase maintainability and portability of the system.

Tools provided by the community helps generate or develop hibernate application very fast and easy. Main architecture is divided into two parts as – web side and engine side. Web configuration is done with struts frame work where model-view architecture is enriching the application server JBOSS. This is handling the client request, session and servers response to the user. Engine side work is being handled by Enterprise Java Beans which pre-processes the source text, translates the pre-processed text and then morph synthesized, collated and ranked.

The users uses the Internet client applications to feed the source document to the EILMT system and get them machine-translated. Expected skill-set of such users is familiarity with PC usage in an office environment i.e. they have some basic computer experience, such as surfing the Internet or using a word processor.

7. SECURITY

Penetration testing web applications is not an easy task. So understanding possible attack scenarios when it comes to securing the web application is obligatory to get a flawless system. To find vulnerabilities in our web application we have undergone a standard for performing application-level security verifications. OWASP is an open community dedicated to enabling organizations to conceive, develop, acquire, operate, and maintain applications that can be trusted. We used this open source web application security project to implement against the top 10 security threats. Servers along with Database are secured with password authentication. Basic password authentication and role based security mechanisms are being used to protect system from unauthorized access. Other security mechanisms are handled by the parent system (i.e. TDIL Data centre).

Web applications present a complex set of security issues for architects, designers, and developers. The most secure and hack-resilient Web applications are those that have been built from the ground up with security in mind. Deploying web based applications over the Internet can be a challenging and intimidating experience. Additional care have to be taken for securing web application from unauthorized access and load unbalancing.

The stateless nature of HTTP means that tracking per-user session state becomes the responsibility of the application. As a precursor to this, the application must be able to identify the user by using some form of authentication. Given that all subsequent authorization decisions are based on the user's identity, it is essential that the authentication process is secure and that the session handling mechanism used to track authenticated users is equally well protected.

- User registration form is been secured with a service known as CAPTCHA, which is a program that protects websites against bots by generating and grading tests that humans can pass but current computer programs cannot. It is a type of challenge-response test used in computing as an attempt to ensure that the response is generated by a person
- Handling Session Management. To secure the communication protocol on informative basis and add efficiency some security related issues were handled such as Injection, Broken Authentication and Session Management, Cross-Site Scripting, Insecure Direct Object References, Security Misconfiguration, Sensitive Data Exposure, Missing Function Level Access Control, Cross-Site Request Forgery, Using components with Known Vulnerabilities,

Invalidated Redirects and Forwards. These aspects are important for authentication, session management and secure consequential data in application which leads our system flawless from vulnerabilities.

- Providing Auditing and Logging functionality which helps to spot the signs of intrusion, inability to prove a user's actions, and difficulties in problem diagnosis.
- Capture of session identifiers resulting in session hijacking and identity spoofing.
- Identifying Security Policies and Procedures which helps Security policy determines what your applications are allowed to do and what the users of the application are permitted to do. More importantly, they define restrictions to determine what applications and users are not allowed to do.
- Use of Connection Pooling which allows multiple users (clients) to make use of a cached set of shared and reusable connection objects providing access to a database. Opening/Closing database connections is an expensive process and hence connection pools improve the performance of execution of commands on a database for which we maintain connection objects in the pool. It facilitates reuse of the same connection object to serve a number of client requests. The application server (JBOSS Server) handles the responsibilities of creating connection objects, adding them to the pool, assigning them to the incoming requests, taking the used connection objects back, returning them back to the pool, etc. When a dynamic web page of the web-based application explicitily creates a connection (using JDBC 2.0 pooling manager interfaces)

8. SCENARIOS

Once the code implementation and Installation is done, system is operated as Web application. Main integrated system with windows operating system are placed on main application server having User log Module, Input Format Extractor, Pre-processing, TAG Parser & Generator, EBMT translation engine, morph synthesizer, post processing, collation and ranking modules. Server 2 is situated with Fedora operating system running SMT engine on it. Clients make a request to main server for translation; main server makes intermediate call to respective engines for translation, collates the outputs, ranks and sent back to user.

9. CONFORMANCE WITH STANDARDS –

System is compatible with W3C Consortium. http://www.w3.org/Consortium/

10. DEPLOYMENT SCENARIO

It is a web based translation system deployed at TDIL-Data center. The JBoss Application Server (AS) is the leading open source J2EE container today. Fully compliant with the J2EE 1.4 specification, it offers a container for Servlets and JSPs; an EJB container for Entity Beans, Session Beans, and Message Driven Beans; Web Services, database connection pooling, and JavaMail support. JBoss Application Server ships with best-of-breed applications including Apache Tomcat for the web tier, Hypersonic for embedded database services, and Hibernate for object-relational mapping. A J2EE certified platform, JBoss Application Server provides a useful tool for developing and deploying Java applications, Web applications and Portals.

11. DEPENDENCIES

EILMT Phase-II shall be able to communicate with the internet communications architecture.

12. EXPANDABILITY

Being in plug and play architecture, system is expandable.

13. DEBUGGING

The system using JBOSS application server which generates the logs of processes being handled onit. The log can be stored and explored for debugging purpose.

14. AVAILABILITY

The system shall be available 24 X 7.

15. PORTABILITY

The web application is coded in J2EE, Struts and hibernate, therefore, it should be transferable between different OS and Java, System can port with any database due to hibernate implementation

16. SALIENT FEATURE

In order to achieve the goal, the system is facilitated with features such as –
- User Log module with
    - User friendly Graphical User Interface
    - New User Registration Module
    - The user can either upload and edit a file/document or type directly into the text area of the system, for translation

- Pre-Processing module to prepare input text into engine suitable form with the help of
    - Input Format Extractor for extracting text from uploaded files and translating for the formats.rtf, .xls, .txt and .html
    - Morphological Analyzer
    - Part of Speech Tagger
    - Named Entity Recognizer including Name, Place, Date, Act & Rules
    - Word Sense Disambiguator
    - Noun/Phrase Chunking, Clause Identification
    -

- The System is designed to use three-translation engines working in parallel namely EBMT, SMT & AG, which would facilitate the translation for all the eight language pairs.

- The Collation & Ranking Module which is responsible for collating translated outputs of all the engines for a given language pair and rank them on the basis of translation accuracy.

- Post processing module provides additional features for EILMT Translation engine like
    - Morph Synthesizer for smoothening the translated output
    - Multiple translation options
    - Synonym selection option
    - Typing facility for Target Languages
    - Transliteration Facility
    - Retaining the original format of English text

- System shall be compatible with W3C Consortium

- Browser compatibility shall be provided for popular browsers such as Google Chrome, Mozilla Firefox, Internet Explorer, Opera and Apple Safari.

17. GLOSSARY

AAI	-	Applied Artificial Intelligence
Client	-	DIT
DIT	-	Department of Information Technology
EBMT	-	Example Based Machine Translation
EILMT	-	English to Indian Languages: Machine Translation
GUI	-	Graphical User Interface
Html	-	Hypertext markup language
IEEE	-	Institute of Electrical and Electronic Engineers
JSP	-	Java Server Page
LRMT	-	Language Resource Management Tool
MO	-	Multiple Outputs
MT	-	Machine Translation
NER	-	Named Entity Recognizer
POS	-	Part of Speech
RAM	-	Random Access Memory
SDD	-	Software Design Document
SRS	-	Software Requirements Specification
SMT	-	Statistical Machine Translation
SSF	-	Shakti Standard Format
STM	-	Statistical Translation Model
TAG	-	Tree Adjoining Grammar
UI	-	User Interface
Web Site	-	A place on the World Wide Web
WSD	-	Word Sense Disambiguation

System is available at -
http://temp-eilmt.cdac.in/eilmt/login.jsp

✡✡✡