

Multilingual Interface for Grid Market Directory Services: An Experience with Supporting Tamil

S.Thamarai Selvi*, Rajkumar Buyya**, M.R. Rajagopalan#, K.Vijayakumar*, G.N.Deepak*

*Department of Information Technology
Anna University, MIT Campus
Chrompet, Chennai – 600044, India
Email : stselvi@annauniv.edu

#Centre for Development of Advanced Computing
Senthil Towers, A-1, First Avenue
Ashok Nagar, Chennai - 600 083, India
Email : mrr@cdacindia.com

**Grid Computing and Distributed Systems Laboratory
Department of Computer Science and Software Engineering
University of Melbourne, Australia
Email : raj@cs.mu.oz.au

Abstract

Providing Internet and Web application interfaces in local languages is gaining popularity recently as it helps in making their services accessible to people who cannot understand English language. Grid computing has emerged as one of the recent applications of Internet and aims at delivering computing as the 5th utility much like the four fundamental utilities—electricity, water, telephone, and gas. This paper presents our work on localization and provisioning of Grid computing services in Tamil (a language spoken in the south Indian state of Tamil Nadu). We have proposed and developed a Tamil interface for the Grid Market Directory (GMD) services as it aims to provide one of the fundamental Grid services that are essential for bringing together producers and consumers of Grid services. The paper presents architecture for location of the GMD and its user interfaces, implementation mechanism, and steps for localization. It then provides snapshots of GMD interfaces in Tamil and demonstrates the overhead of providing GMD services in Tamil is very minimal compared to presenting the same in English.

1. Introduction

Grids [2] are emerging as a global cyber-infrastructure [3] for solving large scale problems in science, engineering and business. They enable the sharing, exchange, discovery, selection and aggregation of geographically distributed, heterogeneous resources such as computers, data sources, visualization devices and scientific instruments. The emergence of grid as a platform for sharing and aggregation of distributed resources increases the need for mechanism that allows an access in local languages. Localization of grid tools has been identified as one of the potential solutions as it helps in enabling the end users to make use of the grid services in local language. Localization of grid computing environment needs to support an infrastructure that enables the creation of user interfaces in local language.

The Grid Market Directory (GMD) [1] developed by the Gridbus Project serves as a registry for publication and discovery of Grid service providers and their services. It enables Grid service providers to publish their services and related costs to the public, so that consumers can browse through web browser or query by using SOAP to find a suitable service meeting their requirements. GMD users use the web services and their own API for creating the Grid services. Grid services support discovery, register and usage. Web services use SOAP (Simple Object Access Protocol), WSDL and UDDI. We use SOAP to send and receive standard messages. WSDL tells how to interact with web services and for client implementation. UDDI gives what are all the web services available and how to register the services. To provide all these services in local language by a common man in the world, localization of GMD is developed. This localization would really be very useful for a common man who is not aware of

International language. Also the localization done is generic, in the way that it can be localized to any of the native languages required.

2. GMD Architecture

The key components of the GMD are GMD Portal Manager (GPM) that facilitates service publication, management and browsing. GPM allows service providers and consumers to use a web browser as a simple graphical client to access the GMD. GMD-Query Web Service (GQWS) enables applications to query the GMD to find a suitable service that meets the job execution requirements. Both the components receive client requests through a HTTP server. Additionally, GMD repository, a database is configured for recording the information of Grid services and service providers.

a. GMD Portal Manager

The architecture of the GMD Portal Manager (GPM) is shown in Fig. 1. The GPM provides three different access interfaces: service browsing, provider administration and service management.

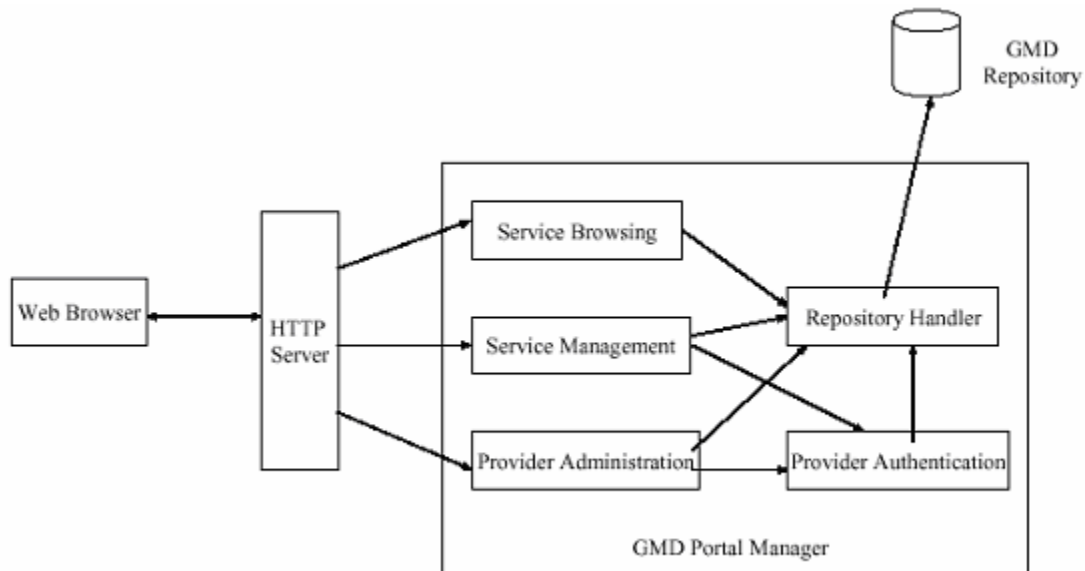


Fig. 1 GMD Portal Manager.

Service browsing

The GPM allows users to browse all registered services or only services offered by a specific provider. Additionally, services in the GMD are categorized by service type, such as Earthquake Engineering, Molecular Docking and CPU Service, so that users can browse them for a particular application area. For instance, the high-energy physics community can browse services related to its area along with their access costs.

Provider administration

The provider administration module is responsible for account management including registration and removal. The account information of the provider is acquired at the time of registration. This includes the provider's name, login name, password, contact address and some additional information.

Service management

The service management module enables the registered providers to maintain their services in the GMD. A service management page is dynamically generated for each registered provider, through which it can add, update and remove services. Basic service attributes include service name, service type, hardware price (cost per CPU-sec), software price (cost per application operation), node host name and location of

application deployment (path). In addition, security issues are also addressed in the GPM. A login authentication mechanism for identifying registered providers is employed in the service management and provider administration. In the service management interface, service modification operations are also authenticated before being committed to the repository.

b. GMD-Query Web Service

The GMD provides web services that applications can use to query the registry. The GQWS is built using SOAP. The main benefit of using SOAP is that it is based on standard XML so that Web services can even be invoked from different applications irrespective of the language used in their implementation.

The GQWS consists of two modules: *Query Processor* and *Repository Handler*. The interaction between the GQWS and its client is illustrated in Figure 5. The GQWS communicates with its client by messages in XML format. The query message is encapsulated within a SOAP message, which is transferred by HTTP between the web server and the GMD client. The SOAP engine acquires the query message and forwards it to the GQWS. The *Query Processor* handles query message parsing and takes appropriate actions based on the content of the message, while the *Repository Handler* is responsible for retrieving data from the database. The response message is finally constructed by the *Query Processor* and sent back to the client.

3. Need for Localization

The ultimate objective of the project is to enable the access of the grid by a common man not having proper knowledge of the International languages. This support is to be provided for rural area people who are lagging their knowledge of International languages like English. This will surely overcome the usage of the grid services to the real world at the present scenario. The translation of all Grid market directory web pages to local languages by the translator designed enables them to provide all the web services in their well known native languages. As they are very familiar with their native languages, the burden for them to get accustomed to these technologies become very easy. All web services translated heir own native language support. This project will surely be a boon to all the rural people in a country like India.

4. Architecture for Localization of GMD User Interface

The architecture as proposed in Fig. 2 clearly states the issues involved at the client side for the localization of the GMD user Interface. The user input is transferred to a local language Translator which inturn maps the corresponding font required for the conversion of the International language to local language. The localized input is then displayed on the Browser and hence the user interface at the client side input is customized for local language support. The HTTP request sent from the browser is then internally translated to International language by another translator which is not visible at the user level as the grid architecture internally understands only the international language for its operation. After the completion of the grid service by the Grid architecture the HTTP response is sent back to the client, which is once again being translated by the local language translator for providing a local language output interface of the service done at the browser of the client side. Hence both the input and the output user level interfaces for Grid services performed are converted for having local language support at the browser on the client machine.

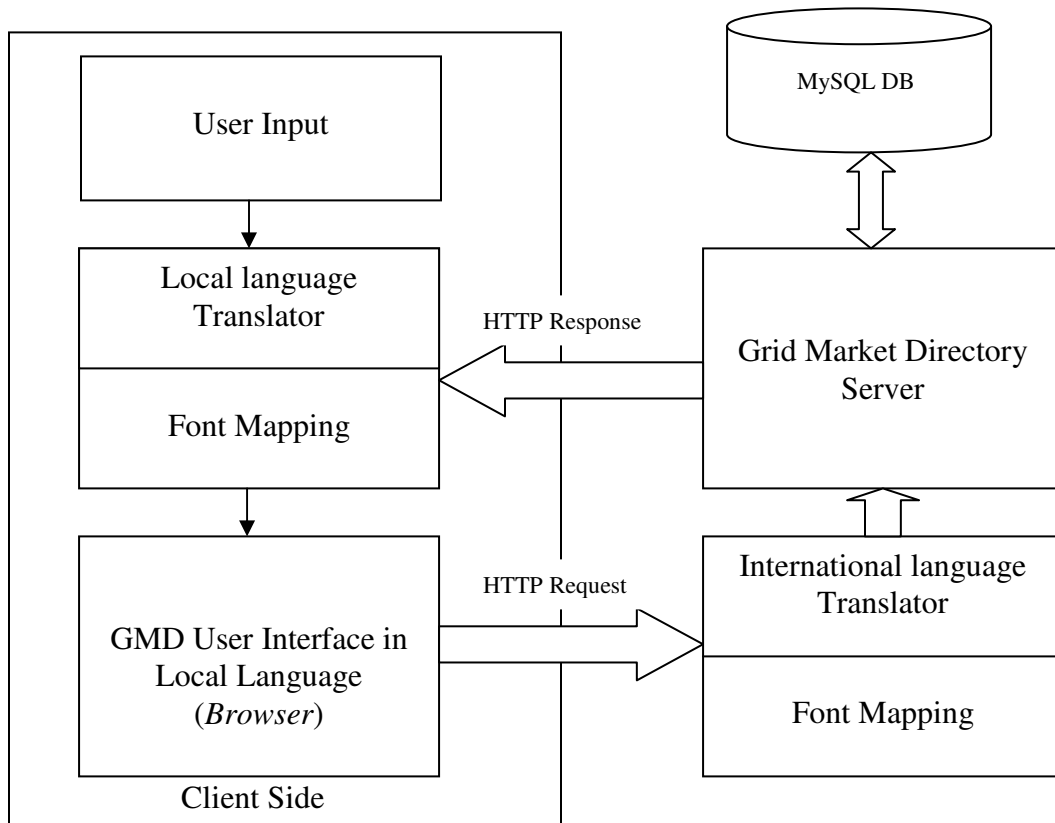


Fig 2: Architecture for localization of GMD user Interface.

5. Implementation of Localization

The grid market directory is modified to support in localization language so that we can give input in localized language, store and retrieve the data in localized language. The grid market directory has four major hyperlinks. They are all services, provider, service type and service management.

All services:

Home page displays all the available services. It specifies the service name, service type, provider name, hardware and software costs, host name and path for all the services. These data are actually retrieved from mysql database. The data part is stored in localized language in mysql and the localized language font is set for the web page so that we can view the data in localized language.

Provider:

This provider link will displays all the provider names. By giving the data in Tamil language and settings the JSP page to Tamil language we can view the names in Tamil. This link also shows two other links, register and delete. Register link is used to register a new provider and delete link is to delete a provider. While registering it asks for username and password. The provider has to use this username and password to add new services or delete existing service.

Service Type:

This is a service type link which shows all the types of services available. This is an JSP page which is also set to Tamil font for the provider name to appear in Tamil.

Service Management:

This link is used to create new services for a specific provider. To create a new service the provider must first give his user name and password to login. After logging in, we get link for creating new services. To create a new service we need to give service name, service type, hardware and software costs, hostname and path. These pages are modified such that we can give the input in Tamil, store the data in Mysql and see them in Tamil language.

To display all the web pages in localized language we used a localized html editor. This editor would allow us to type in localized language. So we can create a web page showing its contents in localized language and also give input in localized language.

Tamil html editor:

We can create a Tamil html using Tamil tags. It parses the Tamil html tags and map them to the corresponding English html tags while saving the html. While reading the source it once again translates the English tags to the respective Tamil tags. Also we have to convert each and every Tamil letter typed within the tags to their corresponding Unicode before saving for the purpose of browser compatibility. So creating Tamil html includes: 1. Tamil html tags creation, 2. Editor creation, 3. Parsing html tags, 4. Tamil-to-english translation, 5. English-to-Tamil translation, and 6. Browser compatibility.

Tamil html tags creation is Tamil equivalent for the html tags need to be created which will be used in the editor. There are more than 100 main tags and around 40 attributes that can be used in html. Parsing html tags is to parse the Tamil html tags and identifying its corresponding English tags for mapping purpose while saving the html and to parse the English tags for mapping to Tamil tags while opening the source. The parsing function processes html input, running a list of html tags, attributes and text. Translation of html tags from Tamil-to-English and vice versa was done using a lookup table. The lookup table is a text file that has the English and their respective Tamil tags.

Tamil html editor features:

The html editor is created in Java which will support Tamil language. It uses Unicode Tamil font Amudham (Typewriter keyboard layout). It supports the following features:

1. The editor itself has its entire elements viz. menu, menu item etc., in Tamil language.
2. The html tags are parsed using Java StringTokenizer and lookup table is a text file that contains the English and corresponding Tamil tags.
3. We use Unicode font it also provides browser compatibility.
4. Allow both Tamil and English character to be used and we can switch between them. A special toggle button is provided for this purpose.
5. Allow us to write Tamil html code, save it and open it.
6. The saved file can be opened in any browser.
7. Allow us to open any html file that is not created by this editor and still displays them in Tamil language.

We use the Tamil html editor to create client side applications (html pages or web pages) in Tamil. The client side applications are designed in such a way that it accepts input in Tamil, displays all services

of grid market directory in Tamil and gives the output in Tamil. The files of GMD are to be modified to provide a local language user interface.

6. Modification of GMD files for Localization in Tamil

The pre-requisite for modification of GMD files for localization in Tamil is listed below.

Step 1: set your choice of Tamil font.

Step 2: set the input style.

Step 3: set the style of textarea.

Step 4. set the td style

Step 5: set the font face for all the table data cells.

For example the following code sets the fonts and styles of our choice

Step 1: ``

Step 2: `<style> INPUT { font-family:amudham; font-size:120%; } </style>`

Step 3: `<style> TEXTAREA { font-family:amudham; font-size:120%; } </style>`

Step 4: `<style> TD { font-family:amudham; } </style>`

Step 5: `<table> table data </table>`

Step 6: All the contents are typed in Tamil using Tamil editor

The following files are modified as given below:

| S.No | File name | Modification done |
|------|------------------------------|--|
| 1. | gmdjsp/addrrecord_detail.jsp | Set the font face of your choice within the <body> tag as given in step1. |
| 2. | gmdjsp/display_all.jsp | Set the font face of your choice within the <body> tag as given in step1. and step 6 |
| 3. | gmdjsp/gmdindex.htm | No modification required |
| 4. | gmdjsp/gmdindex.jsp | No modification required |
| 5. | gmdjsp/gmdmenu.jsp | Set the font face of your choice within the <table> tag as given in step5 and step 6 |
| 6. | gmdjsp/gmdprovidermenu.jsp | Set the font face of your choice within the <table> tag as given in step5 and step 6 |

| | | |
|-----|------------------------------------|---|
| 7. | gmdjsp/gmdservicetypemenu.jsp | Set the font face of your choice within the <body> tag as given in step 4. |
| 8. | gmdjsp/gmdtitle.htm | Set the font face of your choice within the <table> tag as given in step5 and step 6 |
| 9. | gmdjsp/gmdwelcome.htm | Set the font face of your choice within the <table> tag as given in step5 and step 6 |
| 10. | gmdjsp/index.htm | No modification required |
| 11. | gmdjsp/loginerror.jsp | Set the font face of your choice within the <table> tag as given in step5 and step 6 |
| 12. | gmdjsp/managementaddform.jsp | Set the font face of your choice outside of the </head> tag as given in step 2 , step 4 and within the <table> tag as given in step5 and step 6 |
| 13. | gmdjsp/managementservicedetail.jsp | Set the font face of your choice within the <table> tag as given in step5 and step 6 |
| 14. | gmdjsp/managementtitle.jsp | Set the font face of your choice within the <table> tag as given in step5 and step 6 |
| 15. | gmdjsp/providerlogin.htm | Set the font face of your choice outside of the </head> tag as given in step 2 and within the <table> tag as given in step5 and step 6 |
| 16. | gmdjsp/providerlogin.jsp | No modification required |
| 17. | gmdjsp/providerout.jsp | No modification required |
| 18. | gmdjsp/providerservicetable.jsp | Set the font face of your choice within the <table> tag as given in step5 and step 6 |
| 19. | gmdjsp/quitloginerror.jsp | Set the font face of your choice within the <table> tag as given in step5 and step 6 |
| 20. | gmdjsp/quitlogin.htm | Set the font face of your choice outside of the </head> tag as given in step 2 and within the <table> tag as given in step5 and step 6 |
| 21. | gmdjsp/quitlogin.jsp | No modification required |
| 22. | gmdjsp/quitprocess.jsp | Set the font face of your choice within the <table> tag as given in step5 and step 6 |
| 23. | gmdjsp/quitwarning.jsp | Set the font face of your choice outside of the </head> tag as given in step 2 and within the <table> tag as given in step5 and step 6 |
| 24. | gmdjsp/Reg_certificate.htm | Set the font face of your choice outside of the </head> tag as given in step 2 and within the <table> tag as given in step5 and step 6 |
| 25. | gmdjsp/reg.htm | No modification required |
| 26. | gmdjsp/registerform.htm | Set the font face of your choice outside of the </head> tag as given in step 2, step 3 and within the <table> tag as given in step5 and step 6 |
| 27. | gmdjsp/register_detail.jsp | Set the font face of your choice within the <body> tag as given in step1. |
| 28. | gmdjsp/reg.jsp | Set the font face of your choice within the <table> tag as given in step5 and step 6 |
| 29. | gmdjsp/remove_detail.jsp | No modification required |

| | | |
|-----|-------------------------------|--|
| 30. | gmdjsp/result.jsp | No modification required |
| 31. | gmdjsp/servicedetail.jsp | Set the font face of your choice outside of the </head> tag as given in step 4 and within the <table> tag as given in step5 and step 6 |
| 32. | gmdjsp/service_management.htm | Set the font face of your choice within the <table> tag as given in step5 and step 6 |
| 33. | gmdjsp/servicemanagement.jsp | No modification required |
| 34. | gmdjsp/servicetypetable.jsp | Set the font face of your choice outside of the </head> tag as given in step 4 and within the <table> tag as given in step5 and step 6 |
| 35. | gmdjsp/tree.jsp | Set the font face of your choice within the <table> tag as given in step5 and step 6 |
| 36. | gmdjsp/update_detail.jsp | Set the font face of your choice within the <body> tag as given in step1. |

7. Results - Tamil Interface Snapshots and Overhead Analysis

The client side web interface of grid market directory is translated to Tamil language and is displayed on the browser in Tamil. Hence any user who is not conversant with International language might be able to access the GMD service in Tamil.

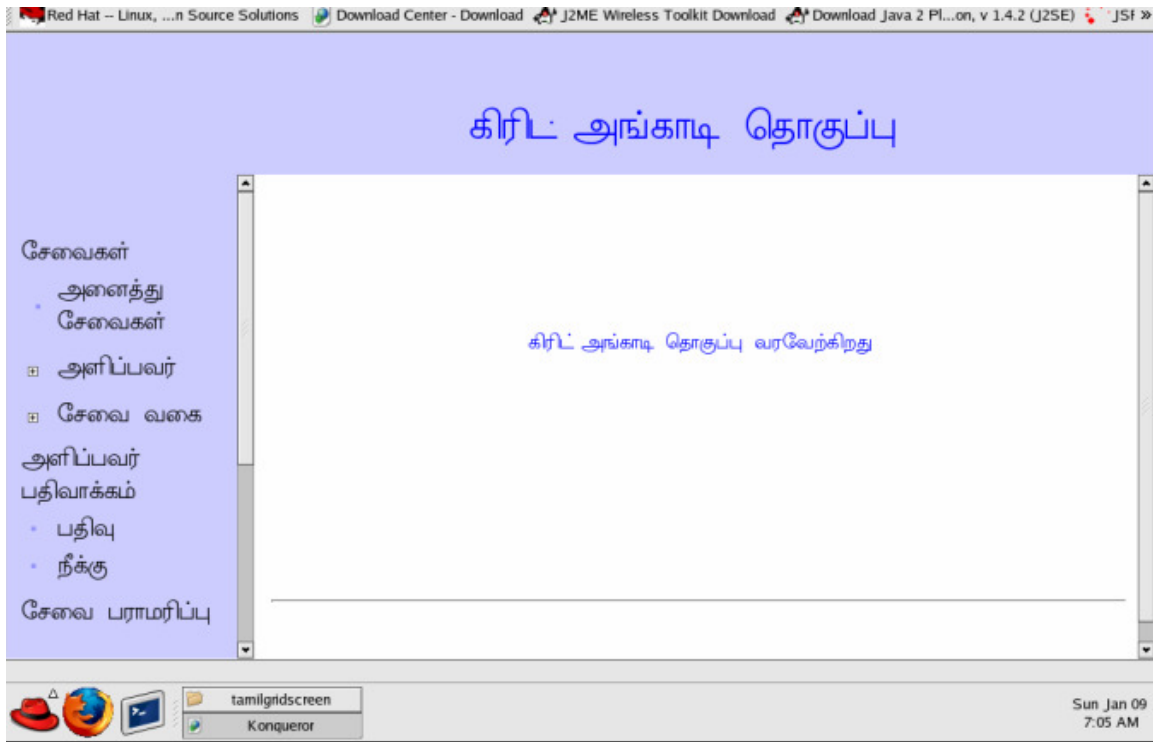


Fig:3 GMD in localization.

Fig.3 shows the localization of the GMD home page for Tamil support. The entire user input and output interfaces translated to facilitate Tamil language support is also clearly portrayed in it.

a. Service Type

கிரிட் அங்காடி தொகுப்பு

அனைத்து சேவைகள்

அளிப்பவர் சேவை

வகை

அளிப்பவர் பதிவாக்கம்

பதிவு

நீக்கு

சேவை பராமரிப்பு

| பெயர் | வகை | அளிப்பவர் | விலை | |
|--------|---------|-----------|----------------------------|-------------------------|
| | | | வன்பொருள் (G\$/CPU-Sec) | மென்பொருள் (G\$/AO*) |
| கிரிட் | குளோபல் | குளோபல் | ரூபி | நகை |

http://10.255.245.222:8080/gmdjsp/registerform.htm

Sun Jan 09 7:06 AM

b. Registration

கிரிட் அங்காடி தொகுப்பு

கிரிட் சேவை

பிறப்பு எண்ரிதழ்

அளிப்பவர் பதிவாக்கம்

பதிவு

அழி

சேவை பராமரிப்பு

பாரத்தை பூர்த்தி செய்து பதிவு பொத்தானை அழுத்தவும்

அளிப்பவர்

பெயர் *

முகவரி

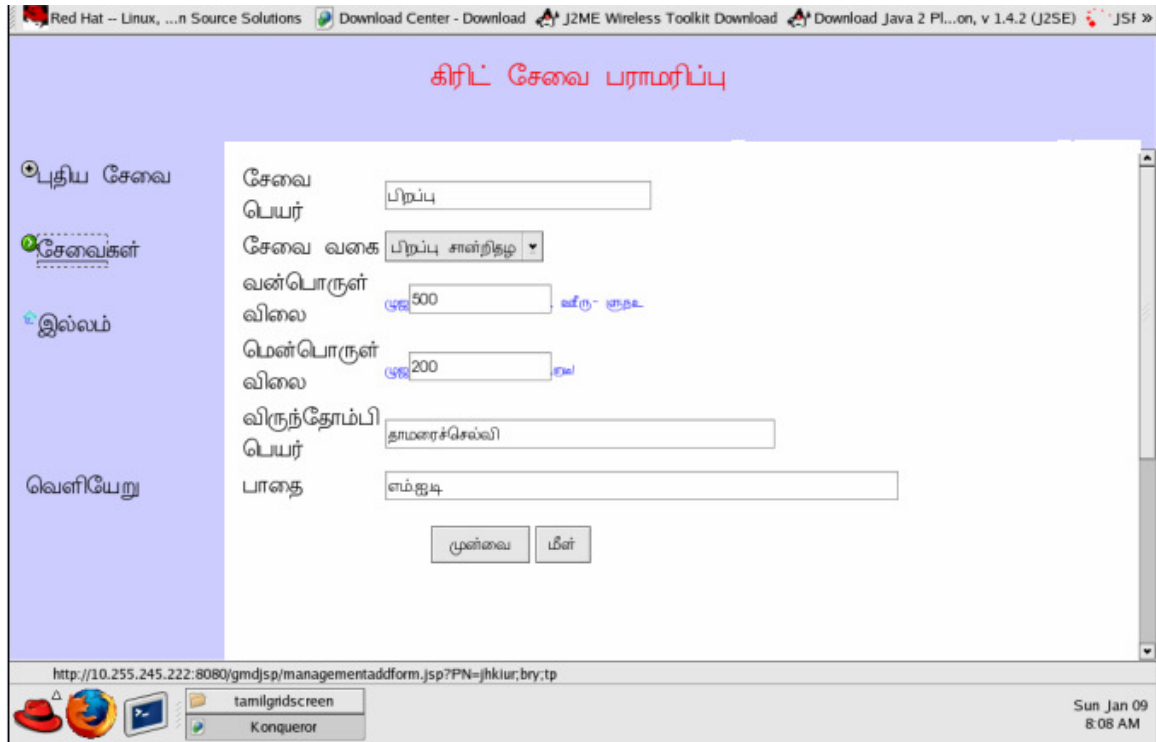
நாடு

பெயர் *

http://10.255.245.222:8080/gmdjsp/registerform.htm

Sun Jan 09 7:14 AM

c. Service Management



d. Localisation Overhead Analysis

In order to evaluate the performance of our project, the execution time of the Tamil GMD is measured. The http response time and request time for both GMD versions in English and Tamil at run-time are calculated and tabulated as shown in Table 1. The major factors that affect the response time are network latency, machine speed and the implementation of http server, SOAP engine and database. We tested the system on a Pentium 4 (1.8GHz) PC. We used Apache tomcat 3.2.4 as http server, Apache SOAP 2.2 as SOAP engine and Mysql 3.23 as database. Whatever the experiments have been done for GMD Services in English can also be done in Tamil without affecting the performance.

Table 1. Average response and request time of GMD Services.

| GMD in Tamil (Sec) | GMD in English (Sec) | Time Difference (Sec) |
|--------------------|----------------------|-----------------------|
| 0.850304 | 0.850296 | 0.000008 |

It may be observed that the response time and request time do not increase much with the implementation of GMD in Tamil. Since the equivalent Tamil word is mapped to the given English word, only a table look-up operation is an additional operation involved. The same translation can be done for any other international Language supported by Unicode. The GMD Services may be provided in any regional language without affecting the performance.

8. Conclusion

The concept of grid computing, web services and grid market directory is briefly discussed. The architecture for localization of the GMD user interface in Tamil is designed and developed. It is implemented by developing Tamil HTML editor and the Translator that maps the corresponding font

required for Tamil support. Thus all the Grid services are translated to Tamil so that one could register, retrieve and publish the information in Tamil. Since the architecture proposed is generic in nature the localization can be achieved for any local language including Chinese and Spanish supported by Unicode. The performance of Tamil GMD services is not much affected since the translation involves only a mapping of an English word to its equivalent Tamil word or vice-versa.

References

- [1]. Jia Yu, Srikumar Venugopal, and Rajkumar Buyya, A Market-Oriented Grid Directory Service for Publication and Discovery of Grid Service Providers and their Services, Technical Report, GRIDS-TR-2003-0, Grid Computing and Distributed Systems (GRIDS) Laboratory, The University of Melbourne, Australia, January 2003.
- [2]. I. Foster and C. Kesselman (editors). *The Grid: Blueprint for a Future Computing Infrastructure*, Morgan Kaufmann Publishers, USA, 1999.
- [3]. D. Atkins et. Al. *Revolutionizing science and engineering through cyberinfrastructure*. Technical Report, National Science Foundation, USA, February 2003.
- [4]. Ahmar Abbas, "Grid Computing: A Practical Guide to Technology and Applications", Charles River Media, March-2004.
- [5]. Joshy Joseph, Craig Fellenstein, "Grid Computing", Prentice Hall PTR, December – 2003.
- [6]. R. Buyya, *Economic-based Distributed Resource Management and Scheduling for Grid Computing*, PhD Thesis, Monash University, Melbourne, Australia, April 12, 2002.
- [7]. R. Buyya, D. Abramson, and J. Giddy, *A Case for Economy Grid Architecture for Service-Oriented Grid Computing*, Proceedings of the International Parallel and Distributed Processing Symposium: 10th IEEE International Heterogeneous Computing Workshop (HCW 2001), April 23, 2001, San Francisco, California, USA, IEEE CS Press, USA, 2001.
- [8]. The Grid Market Directory: <http://www.gridbus.org/gmd/index.htm>
- [9]. Marty Hall, *Core Servlets and Java Server Pages*, Prentice Hall PTR, 2001.
- [10]. Eric Ladd and Jim o'Donnel, "Using HTML 3.2", Platinum Edition, Dec- 2001.
- [11]. R. Buyya and S. Vazhkudai, Compute power market: Towards a market-oriented grid. First IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2001), Brisbane, Australia, May 15-18,2001.
- [12]. Jini Technology, <http://www.sun.com/software/jini/>
- [13]. W3C. Web services <http://www.w3.org/2002/ws/>
- [14]. Simple Object Access Protocol (SOAP) 1.1, <http://www.w3.org/TR/SOAP/>
- [15]. W3C, Extensible Markup Language(XML), <http://www.w3c.org/XML>
- [16]. Tomcat open-source servlet container, <http://jakarta.apache.org/tomcat>
- [17]. Java Apache SOAP, <http://xml.apache.org/soap/>
- [18]. JDOM API, <http://www.jdom.org/>
- [19]. MySQL open-source Database, <http://www.mysql.com>
- [20]. Java Server Page Technology, <http://java.sun.com/products/jsp>
- [21]. JDBC Technology, <http://java.sun.com/products/jdbc/>
- [22]. Java Servlet Technology, <http://java.sun.com/products/servlet>