Many modern web sites are built using a conglomeration of technologies. Take text from a database, programmatically generate other text, merge in some static HTML or XML pages, perhaps blend in some configuration data, shake well and serve the resulting pages up

If these pages are to be translated in multiple languages, the traditional approach is to translate each component, the individual text fragments in the database and in the generator program, the static pages, etc. and create a complex system to assemble the pieces while insuring they are linguistically correct. For sites supporting large numbers of pages in many languages, the problem can become intractable

Jujitsu is a martial art. The main concept is yielding and using the opponents energy to your advantage. This paper will propose an alternative approach to globalizing web applications. Rather than burdening the process of creating web pages with the complexity of translating fragments of text and establishing how to assemble them so they are also linguistically correct, our proposal is to leverage the final output, in the original source language, and use it to our advantage to dynamically translate the assembled page. **Jujitsu!**

This greatly simplifies the translation and content management problems of the traditional approach.By using proxy servers, the source language pages are easily intercepted and replaced with their equivalent translations, while strong performance and flexibility are achieved. Schedule conflicts between content developer, software developer, and translators are also greatly reduced

We will show how this method can work, the issues and benefits of this approach, and provide a demonstration

This session will be of interest to anyone either involved with the globalization of web applications or having a business need to reduce the cost and complexity of running a multilingual web site.

We have simple goals for this session. We will describe some of the challenges of localizing typical web sites and describe an innovative solution. We will provide a demonstration of the new approach.

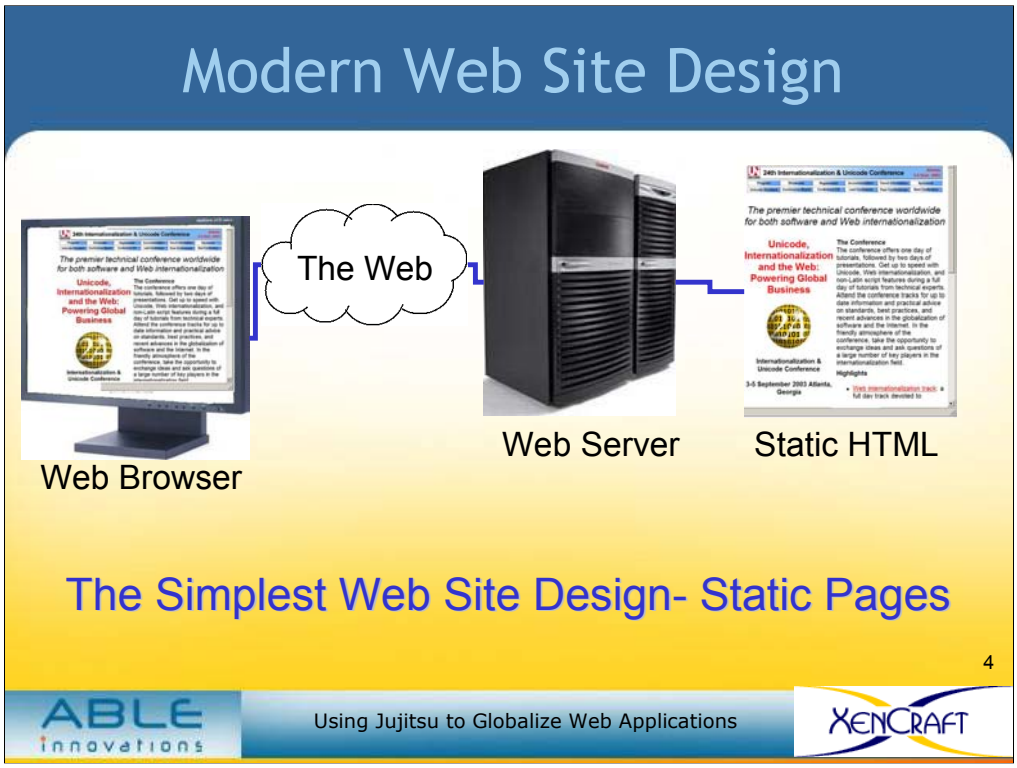
The agenda is very straight forward. We will describe the current approach to building web sites and the inherent localization issues.

We will explain the reference to the martial art Jujitsu.

We will then describe an innovative alternative approach to delivering localized web sites.

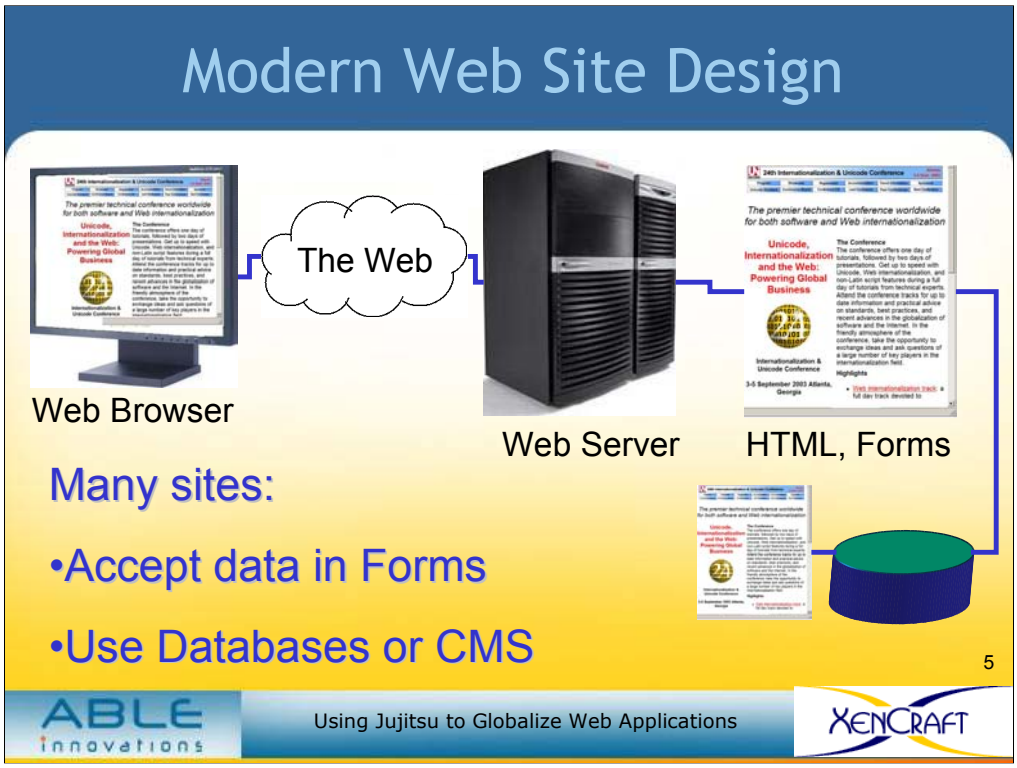
To the extent time allows we will demonstrate the new technique.

Let's begin by discussing web site design.



The simplest of web sites is one that consists of just static HTML pages. It is very effective for publishing information. Because of its simplicity is also easy to localize and configure for different languages. Typically, parallel directories contain translations of the original web pages.

As the site gets large or if the pages are frequently updated, it can become difficult to maintain links and to keep the translations synchronized with the original pages (i.e. insuring that when page A1 is updated to A2, that A1's translation is also updated to reflect a translation of A2).

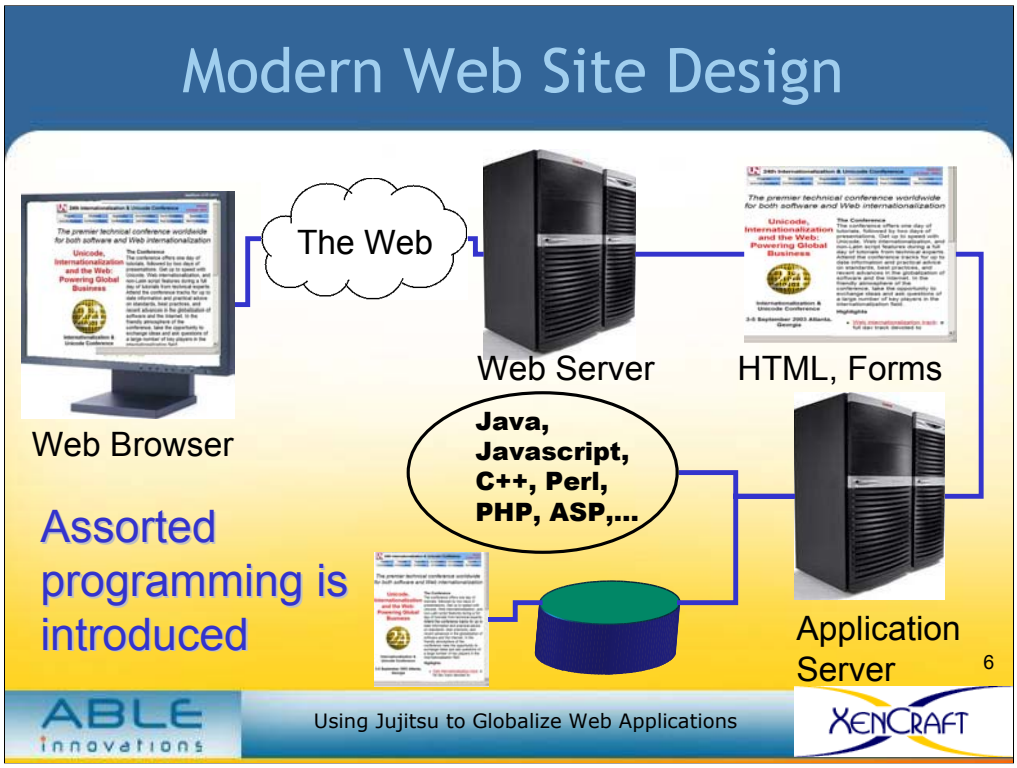


This diagram represents a more typical web site for a business. In addition to publishing information, they have a need to accept data from users. This means posting forms and recording responses in a database.

In addition, companies often have so much product, industry and other information that they want to make available on the web, that the information is stored in a database and used to generate many web pages. The pages may be generated once and stored statically on the server. However, it is also common to generate pages from the database on an as-needed basis. This makes data in the database instantaneously available on the web, as soon as the database is updated.

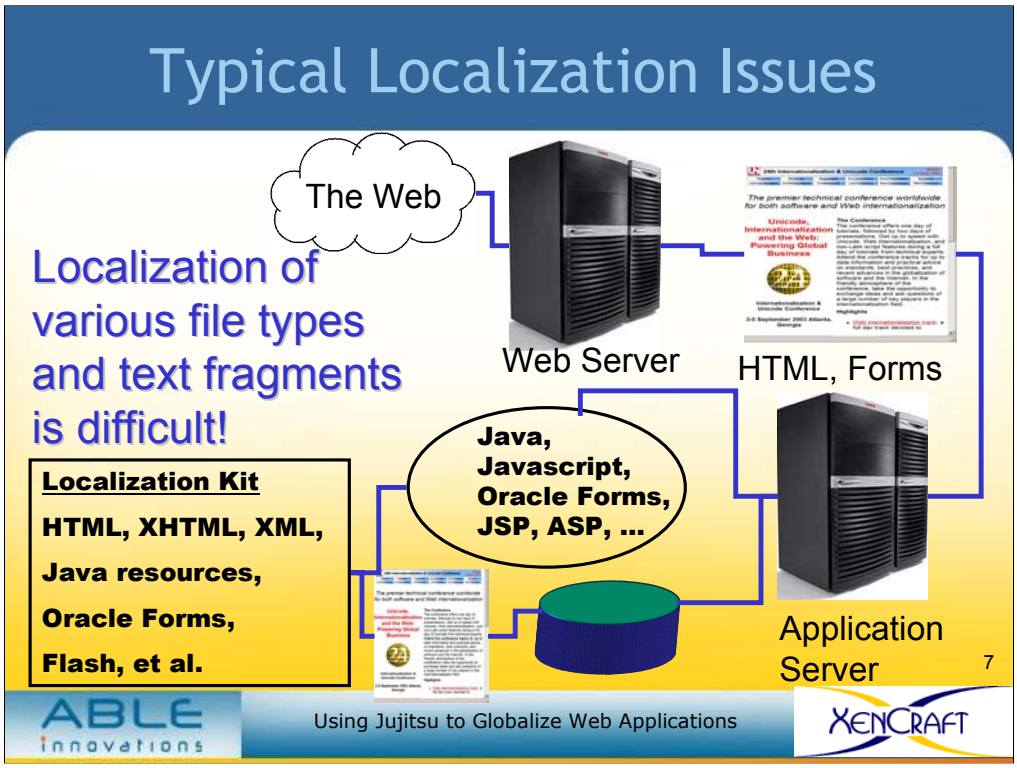
Another common approach, as the number of static web pages gets large, is to store the pages in a database or repository, called a Content Management System (CMS). Sometimes a CMS is specialized for global web sites and is known as a Globalization Management System (GMS).

CMS also offer abilities for managing and tracking workflow, so that new pages can be cycled to localizers, editors, various reviewers, and finally release to the web site.



For many reasons, programming is introduced to the web site. It may be needed to process data returned from users. It can be used to accomplish special effects on the web page. Often data in a database is not in a textual form or in a form directly usable on the web and programming is used to transformed or format the data.

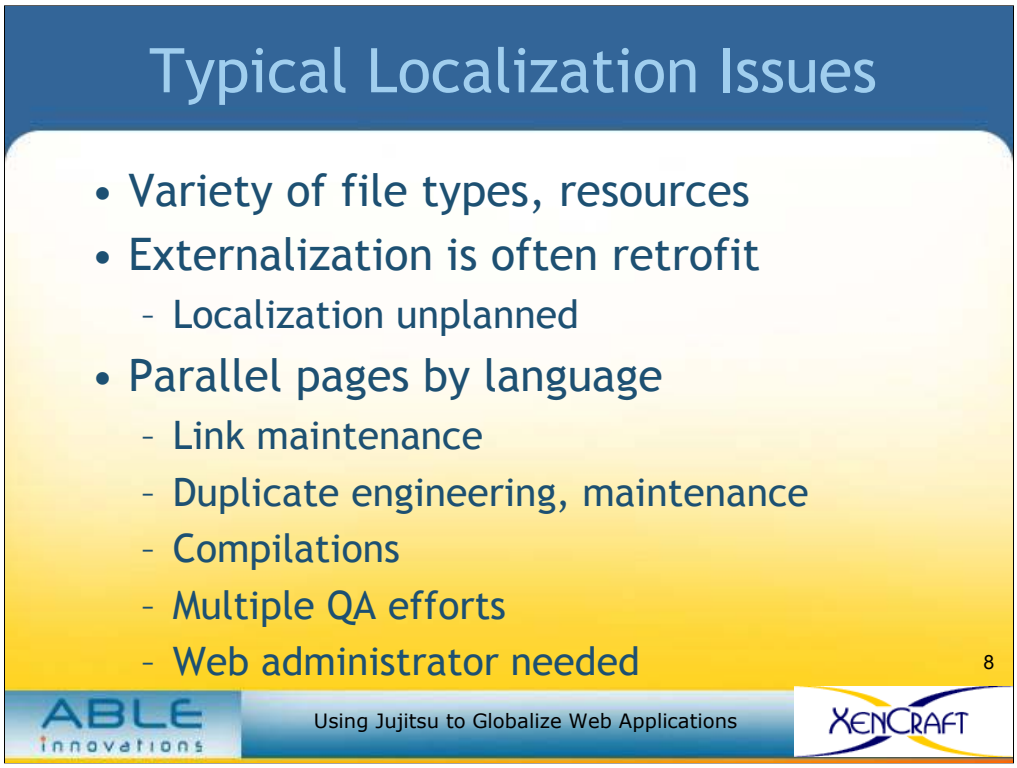
Often, programming is used for dynamic assembly and generation of pages, to provide users information specifically chosen and configured to meet their needs. Besides the benefit to users, it also allows the information to be stored efficiently, since text fragments are not redundantly stored in different text pages. Instead, the fragment is stored once and reused in many different dynamically generated pages.



One issue with modern web sites, is that the information that is to be localized may be distributed in many places and file types. To prepare a kit for localization, we might need to gather files in HTML, XHTML, XML, text, Java resource, Java property, C or C++ resource files, Unix Message catalogs, Oracle forms, Flash files, Graphics files, database contents and others.

Depending on how the text is fragmented and reconstituted, it can be problematic to translate and know how the text will be used. I.E. to know the space available, the semantic context and the grammatical relationship to associated text.

The problem is often exacerbated by the fact that many web sites are first designed without planning for internationalization or localization. It can be difficult and expensive to retrofit the requirements of internationalization and localization into a system that has not planned for it. For example, if the text is not externalized from the programming, there can be a sizable effort to extract and separate the text from code. Depending on how the text is manipulated and concatenated, some reprogramming may be needed so that translated text is pieced together sensibly.



Some of the biggest difficulties, expenses and frustrations resulting from localized web sites have to do with the requirements for Quality Assuring the localization of the site.

As pages are duplicated for each language offered by the site, the underlying engineering and programming is repeated. Engineering changes for site maintenance, made to the original language pages, must also be made to the localized pages. The additional engineering entails some risk of being made incorrectly.

There is also a risk that changes occurring during localization may break the functionality of the site.

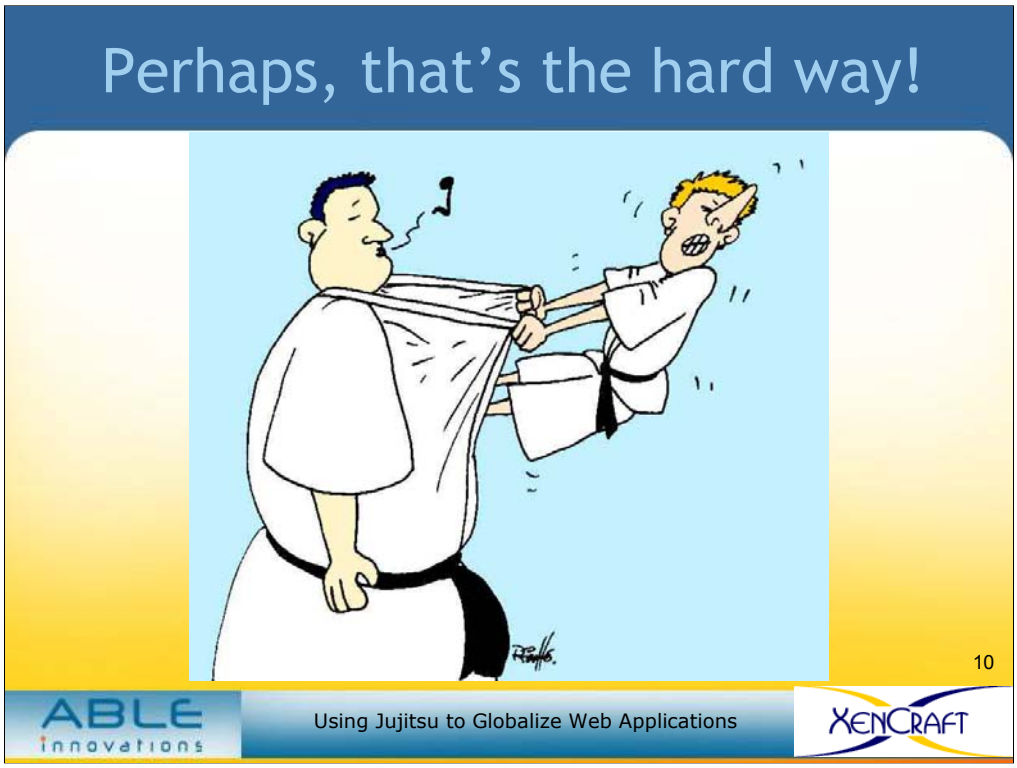
In addition to these risks, the effort to build and run the site is increased, as the number of pages and amount of functionality in the site, is multiplied by the number of languages offered. This results in additional costs for computing resources and to expand the programming, QA, and web administration staff.

Of course, if files with specialized formats are to be localized, the localizers must have skills and/or tools in support of those formats.

For example, a translator skilled with markup languages such as HTML or XML, is more expensive than one that is used to working with plain text files only.

Specialized file formats tend to be more error prone, as they have formatting or validation requirements that plain text files do not have.

Often additional staff is required. In addition to translation. there may be web site engineering, or page layout in the localized language, and then linguistic and functional QA. This increases the cost of localization.



OK, so those are some of the difficulties facing companies that have large sites and needing to support several languages. Is there another way?

What is Jujitsu? You might think from the title of this presentation, that it is a Web site globalization product. No. Jujitsu is a martial art.

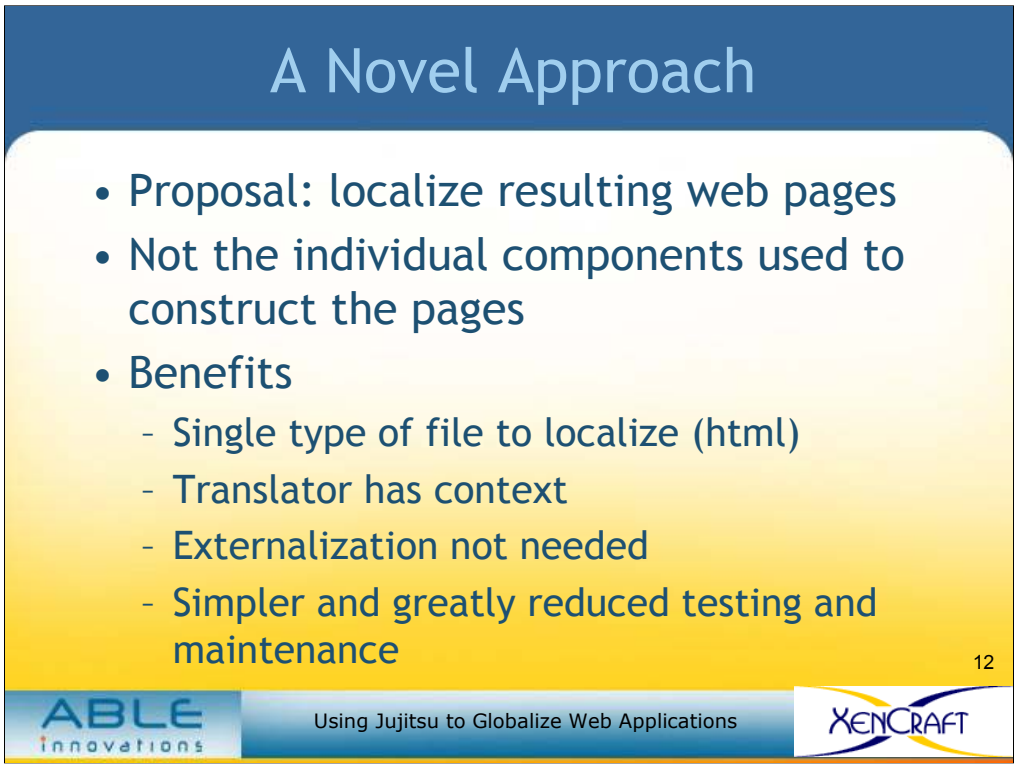
The following description is from http://akayama.topcities.com/history.htm:

'While the most popular translation of jujitsu remains "the gentle art," a more apt translation would be "the art of flexible adaptation". Jujitsu requires the ability to yield or flow with an attack or offer momentary resistance in order to break the attacker's balance and/or momentum and thereby control, disable, cripple, or kill the opponent. True jujitsu is achieving the maximum effect with the minimum effort.'

The point of interest for us, is that Jujitsu leverages the opponent's size and momentum to advantage. If we consider the problems of localization as our opponent, maybe we can somehow use the attributes of large multilingual web sites to our advantage.

By the way, the martial arts graphic on this page and used throughout this presentation comes courtesy of:

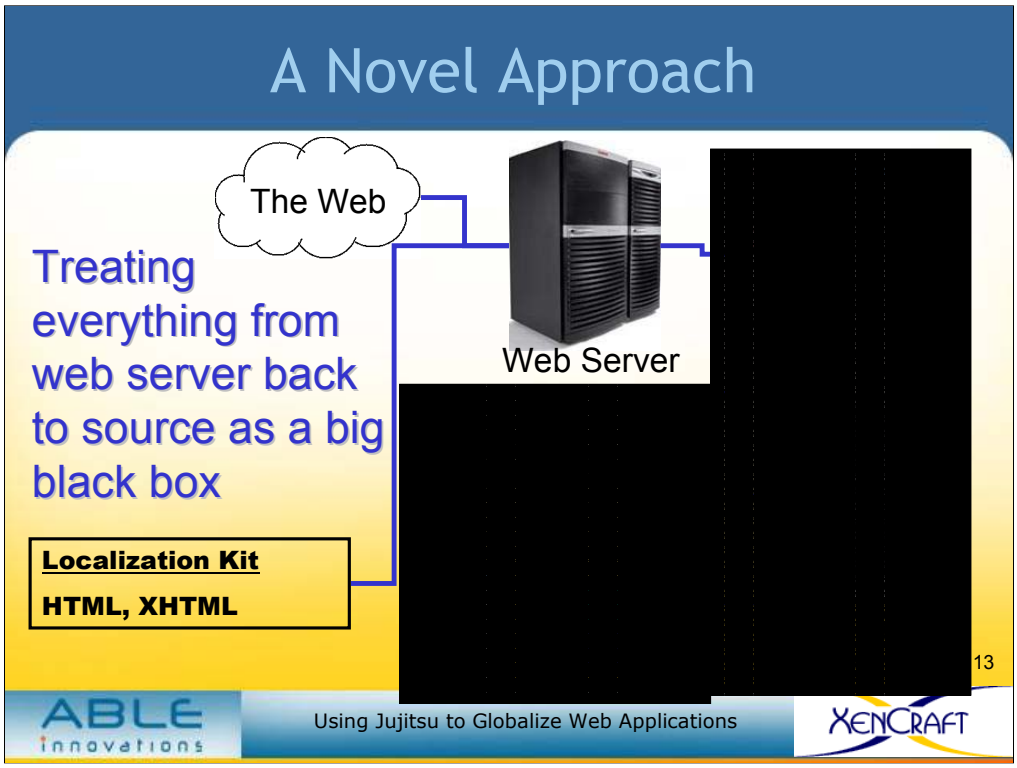
http://www.piedmontymca.org/judo.htm



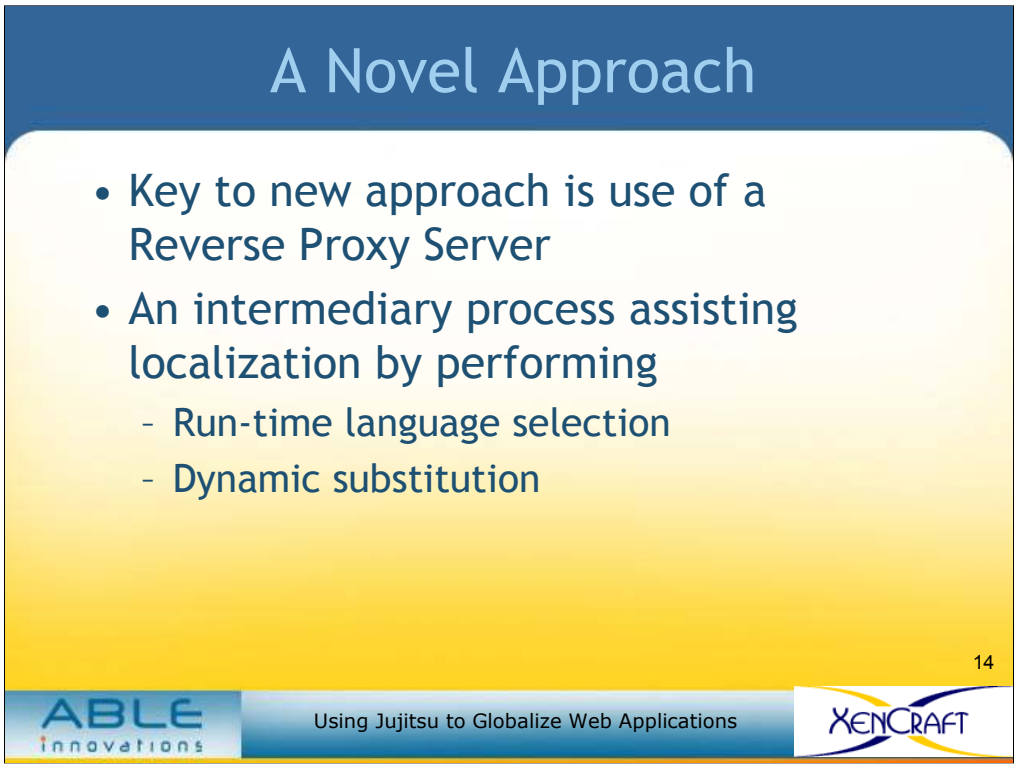
So what does our opponent have that we can leverage and use to our advantage? Well, there are two key elements we might like to utilize. One is the fact that looking from the outside, the website has completely composed pages. If there is a lot of work in dealing with individual components, different programming languages, markup languages and file formats, maybe there is something to be said for dealing with just the end result.

The other key element is that it is on a web server. It is possible to intercede between a web server and its end-user. That gives us a point of access to work on the resulting pages before the user receives them.

The benefits of working with the resulting web pages on the web server are listed on the slide.

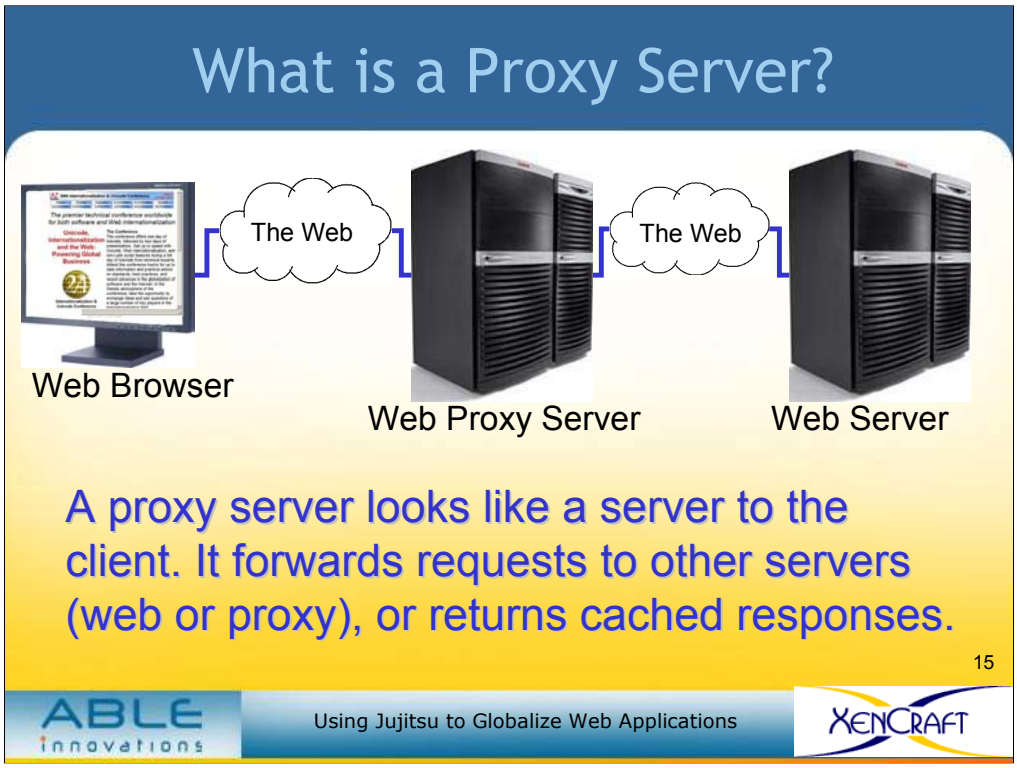


So one idea is to ignore the makeup of the system behind the web server. We will look at localizing the resulting web pages. The localization kit is then simplified since it only contains the resulting web pages. The kit will have HTML, perhaps XHTML, and graphics files and the like.



How can we intercede between the web server and the user? We will use a Reverse Proxy Server. I will tell you more about how this works in a minute.

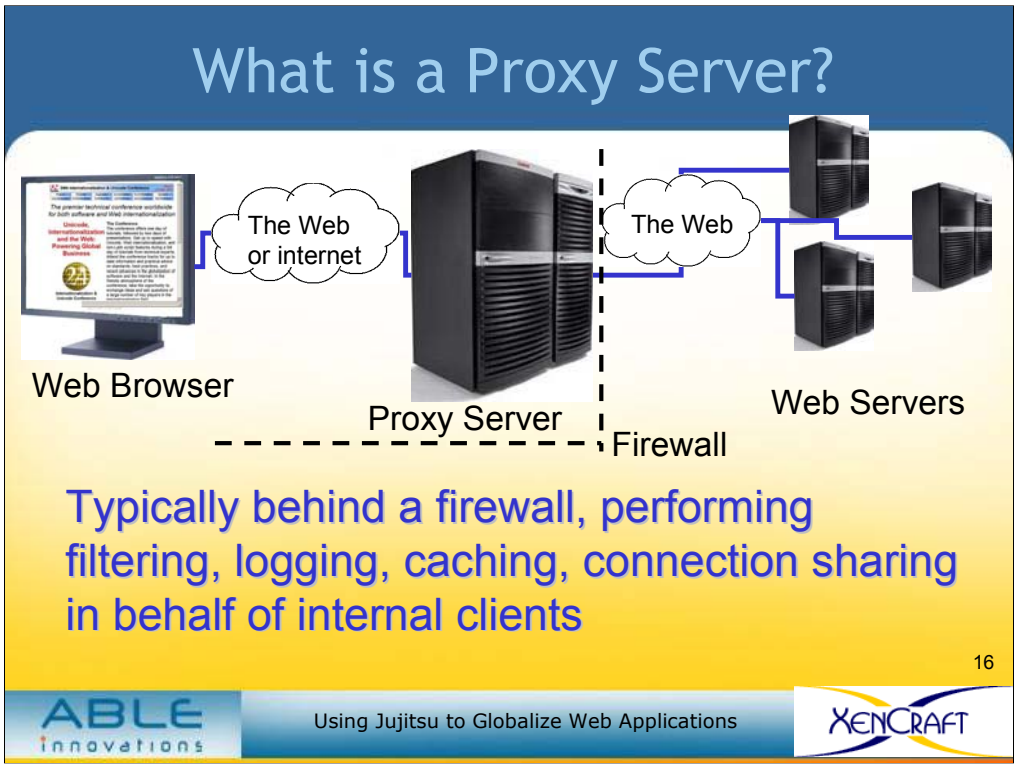
The important point is that this special server will sit between the web server and the user, and accept the web server's web pages, and dynamically substitute localized pages for the original language pages. The Reverse Proxy Server can support multiple languages, and the language it gives to any user can be changed at run-time.



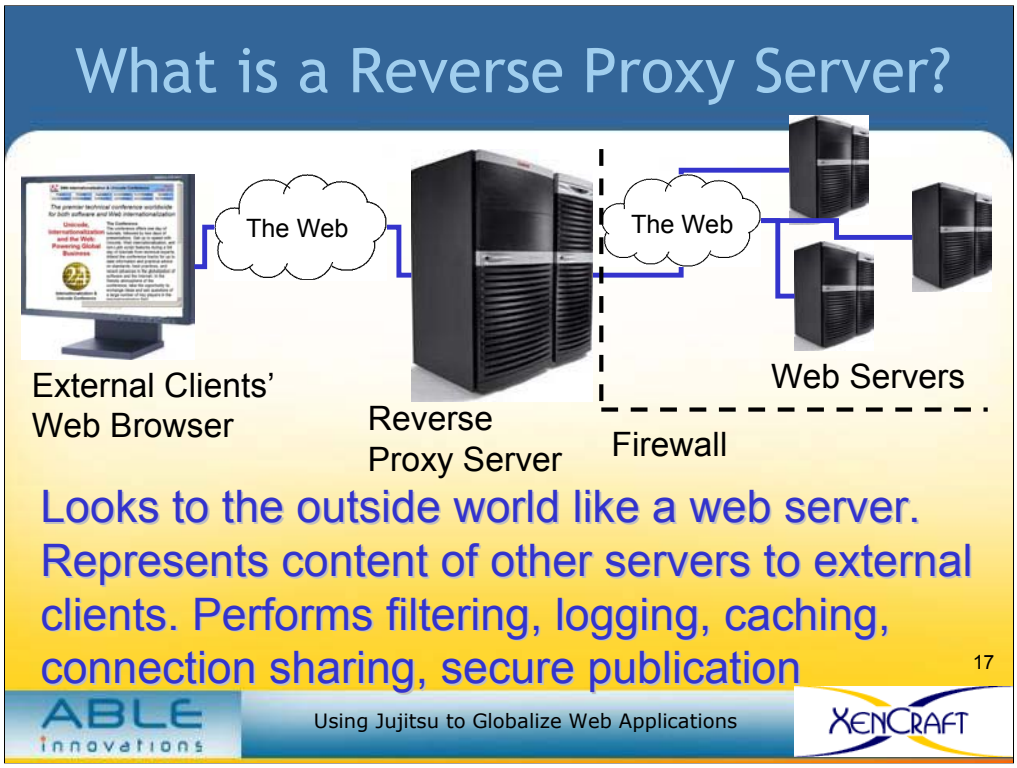
Before we describe the Reverse Proxy Server, let's discuss a simpler concept and one that more people are familiar with, a Proxy Server.

A proxy server looks like a web server to the client. Clients send requests for web pages to the proxy server, and the proxy server in turn redirects the requests to other servers. The proxy server receives the web pages from the servers and then sends them to the requesting client. The Proxy Server can also cache pages so that it can quickly return pages that are frequently requested by its clients.

Proxy Servers can also make modifications to the client requests or to the returned pages. They can also filter requests, log them, and perform other services.



A typical application is for a company to place a proxy server behind a firewall. All employee clients go thru the proxy server to access the internet. This enables capabilities such as filtering, logging, caching and connection sharing to occur. For example, access to inappropriate sites can be prevented.



A Reverse Proxy Server is similar, but instead of representing a set of clients to the internet, it represents a set of servers to the internet. It can perform a similar set of services for the servers. In addition, it can allow for secure publication by restricting access to web pages. This can also be useful in controlling whether localized pages that are works in progress, are seen by external users.

The benefits of Reverse Proxy Servers include that they can accelerate response time and can conserve resources.

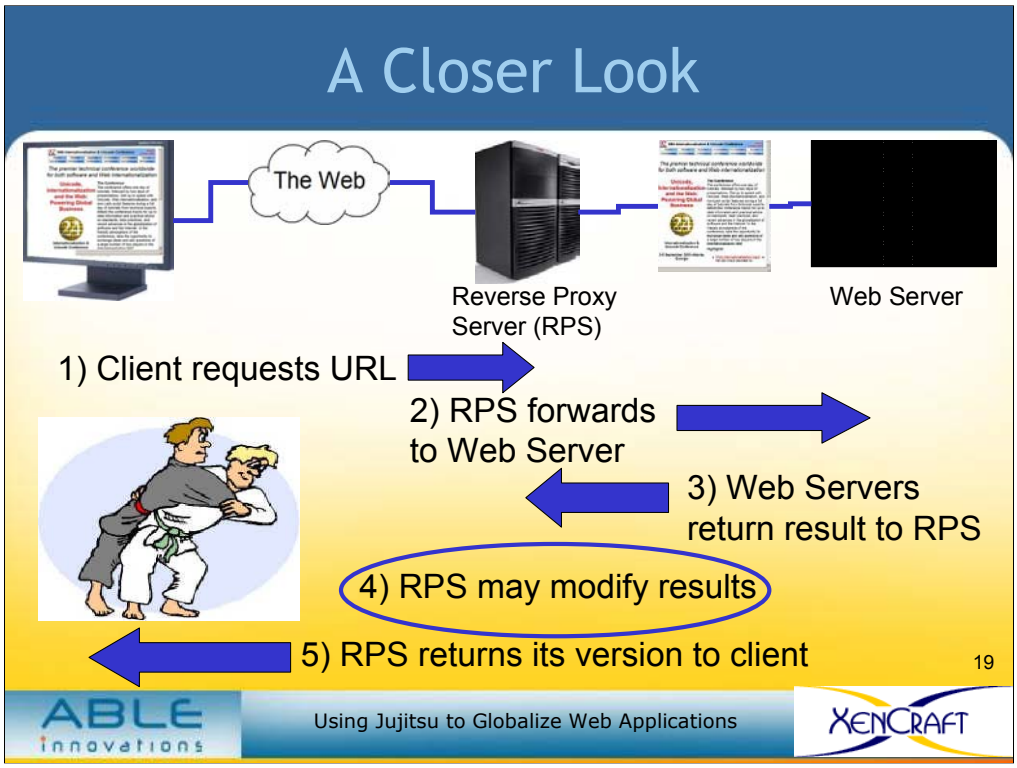
You can read more about Reverse Proxy Servers here:

http://wwws.sun.com/software/products/web_proxy/ds_web_proxy.html

http://vms.process.com/~help/helpproxy.html#E21E46

http://www.webopedia.com/TERM/P/proxy_server.html

So, to reiterate, suppose we use a Reverse Proxy Server to represent our web site to the external world. If it can perform the localization function we want, perhaps it can greatly reduce the work and cost involved.



Here is a closer look at how the Reverse Proxy Server (RPS) would work.

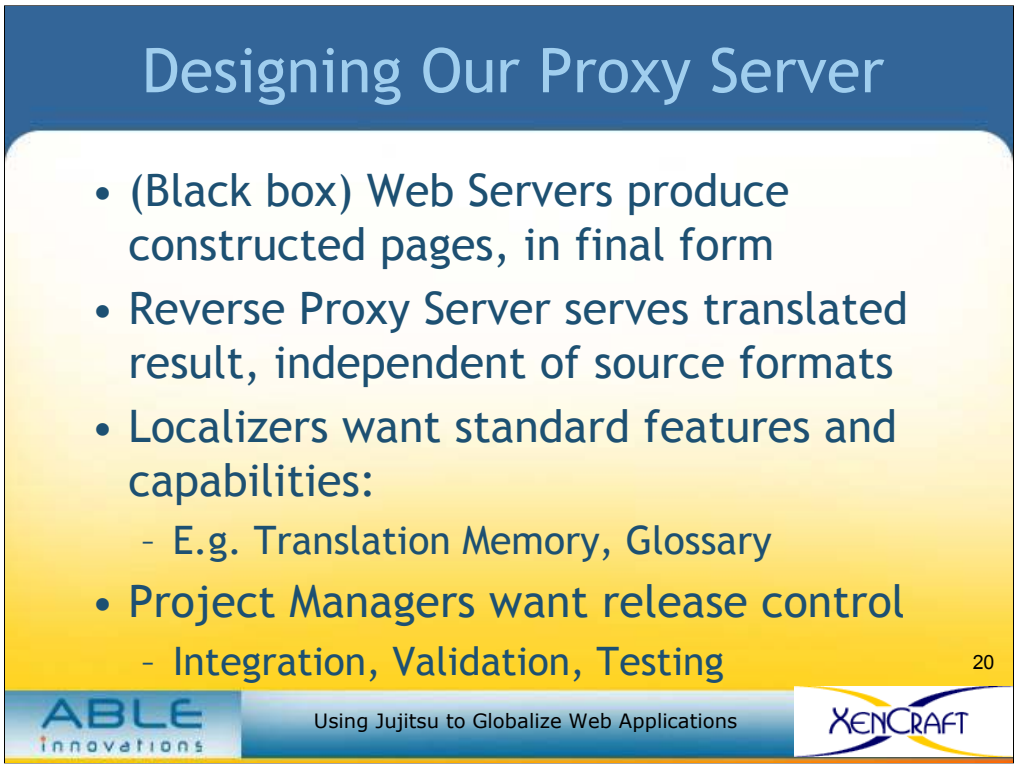
The client would be given the address of our web site. The client would send an HTTP GET request to that address (URL). From the client's perspective, it does not know if it is addressing a web server or the Reverse Proxy Server.

In this case, the address is that of our Reverse proxy Server. The Reverse proxy Server in turn sends the request to the Web Server. The Web Server returns the page to the Reverse proxy Server.

The Reverse Proxy Server may choose to modify the page or perform other actions (e.g. logging, caching, etc.)

Finally, the Reverse Proxy Server returns something to the client. It can return of course the original page that the Web Server gave it, or it can return a modified (e.g. localized) page.

The client has no way of knowing that it was getting a page from the Reverse Proxy Server instead of the Web Server.

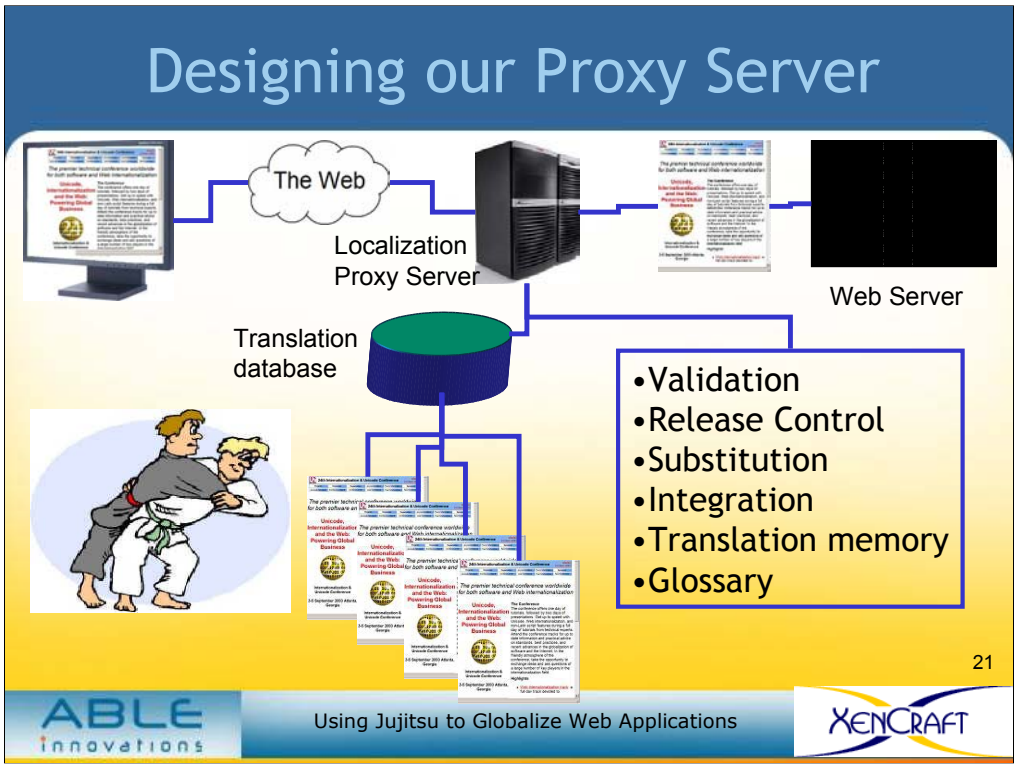


Let's consider our design requirements for our Localization Reverse Proxy Server.

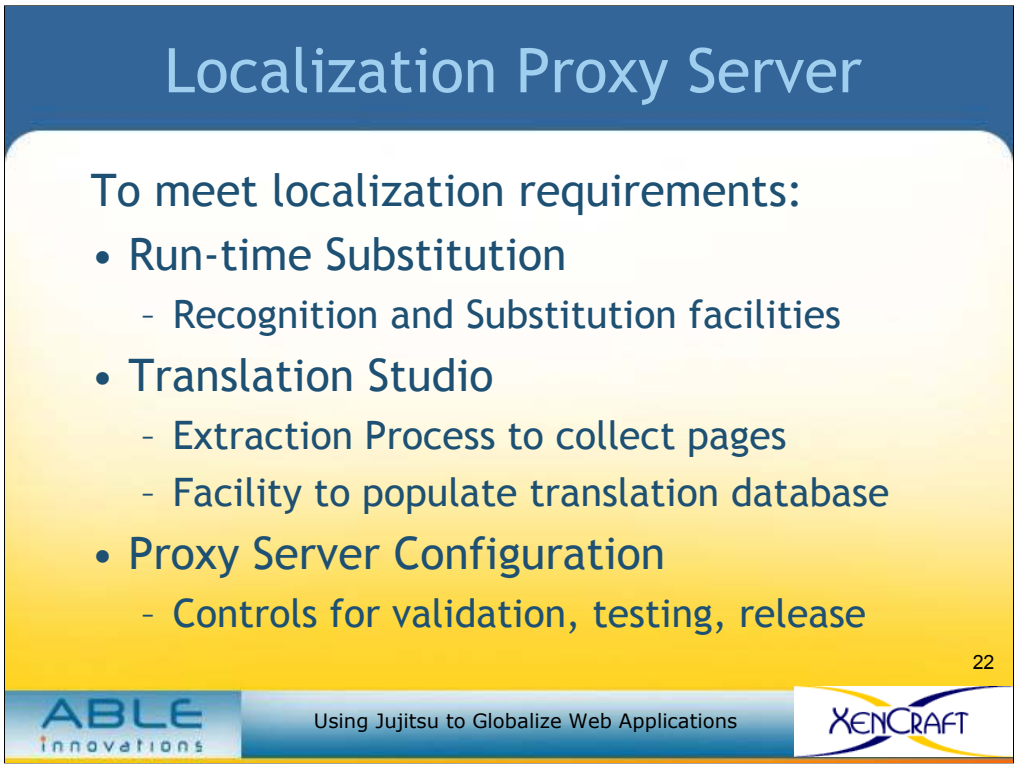
We expect it to receive web pages from a Web Server. We expect it to somehow map these web pages to localized equivalent pages.

The facilities for localization should be industry standard so that localizers do not require special training or tools. They will want features like translation memory and glossary support for reasons of efficiency.

Project managers will want release control, so that pages that are not yet completely localized cannot be seen by external users. However, they should be available to localizers so they can see the results of their handiwork, even if they are working remotely around the world. It will also facilitate testing if testers can receive the web pages from the proxy server, while external users are prevented access.



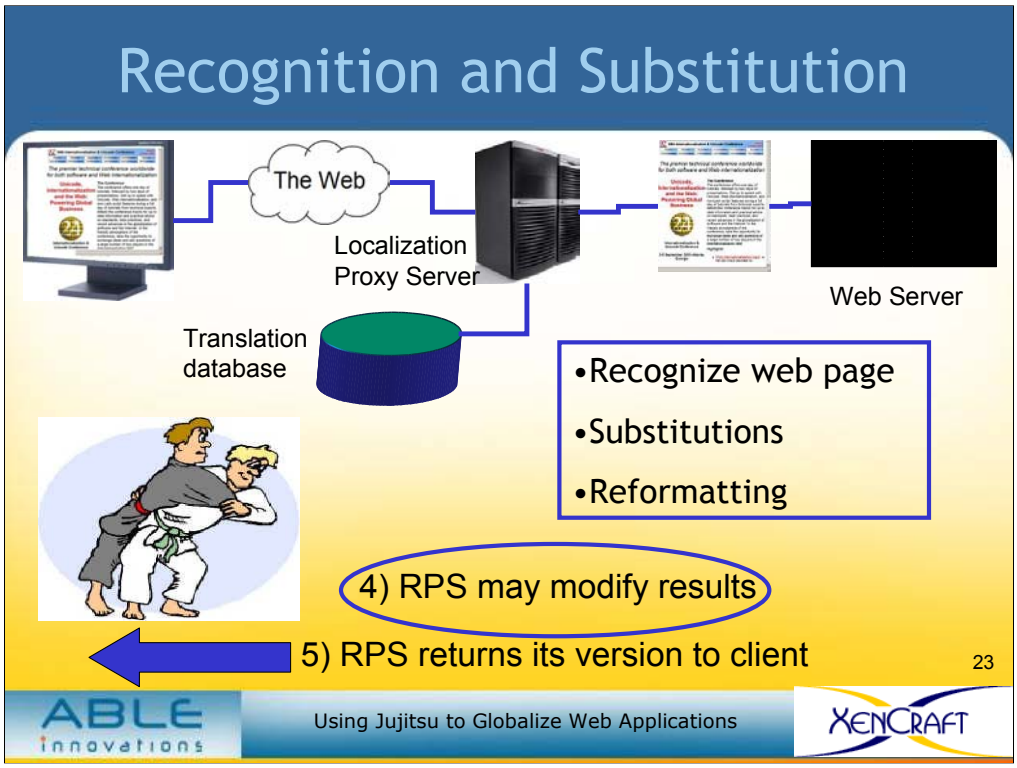
To support storage and retrieval of the localized web pages we will want to associate a database with Reverse Proxy Server, that will provide fast access and search capability.



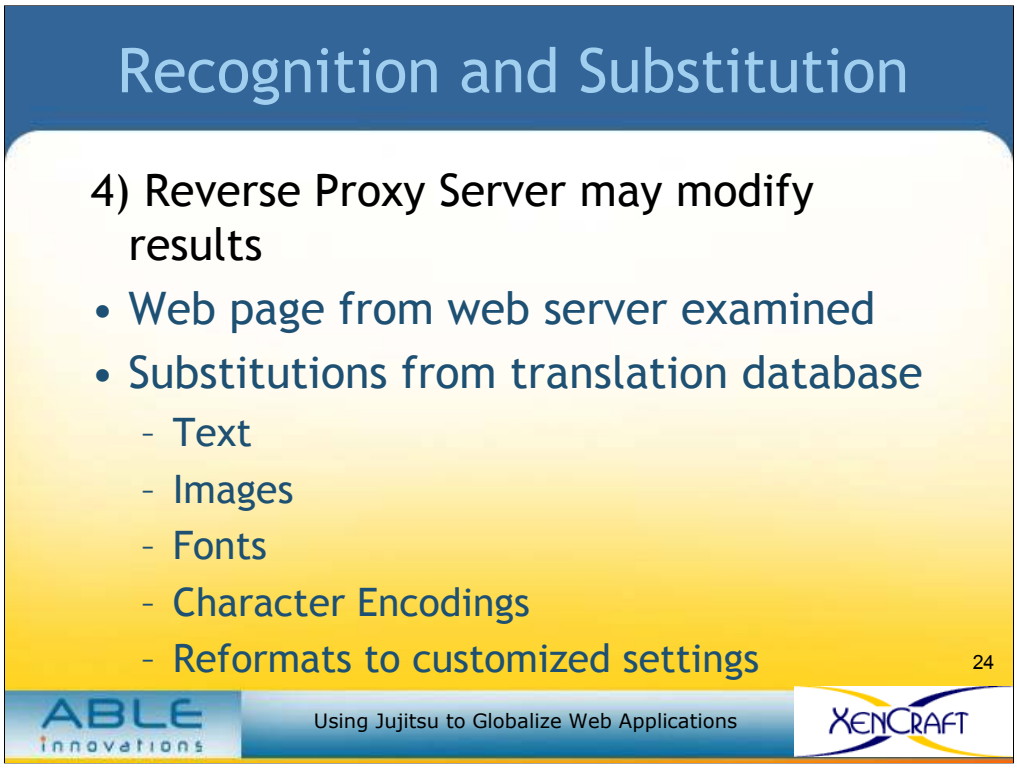
Further, this Localization Proxy Server will need to have the ability to:

1) Collect all of the Web Server's pages
2) Parse the pages, extracting strings, etc. for localization
3) Populate the database with the original strings, graphics files, etc.
4) Support a localization process, populating the database with different language versions of the original contents.
5) Provide access management and lifecycle support for the contents
6) And of course to be able to quickly examine the pages that are in the original language and returned by the Web Server, recognize the page and perform appropriate substitutions of either the entire page, or to parse the page and make appropriate substitutions for parts of it, to create a localized page and return that page to the client.
7) It would also be useful if the Localization Proxy Server could provide a run-time language selection service, to standardize and simplify the choosing of a language by users.

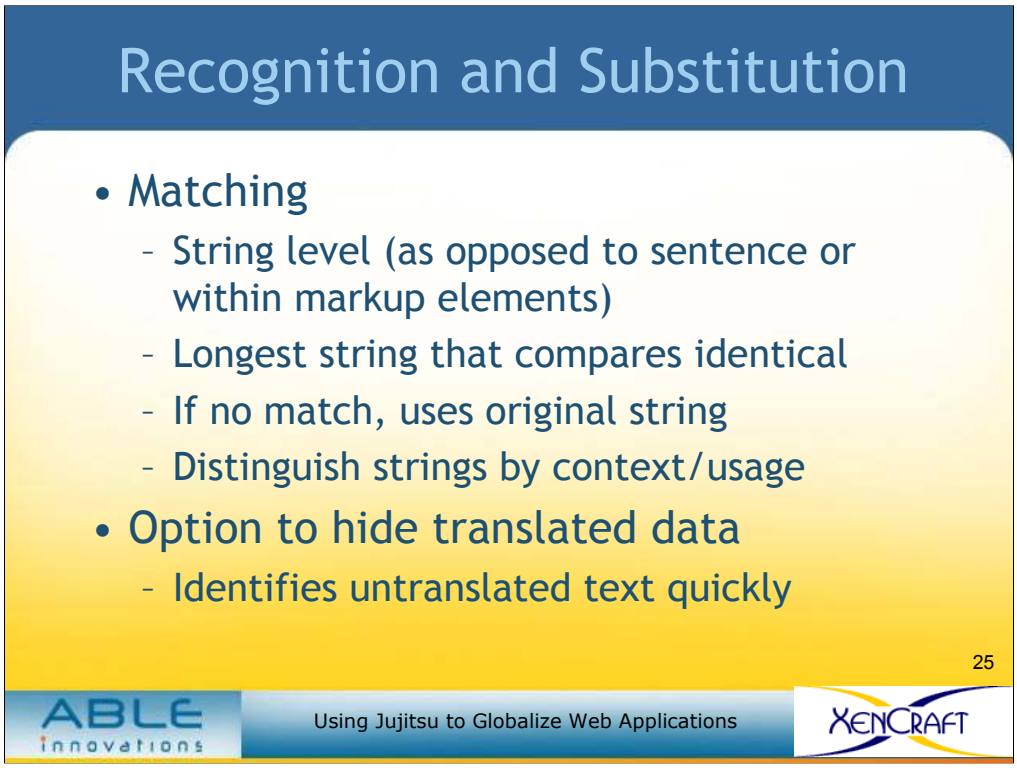
These can be categorized into 3 functionality groups: A **translation studio** which prepares the server to perform translation, the actual **runtime substitution service**, and **configuring of the proxy services**, which determines who has access to the Localization Proxy Server facilities and content.



Let's take a closer look at step 4 and the modifications that the Localization Proxy Server might be capable of.



Here are some of the elements of web pages, for which substitutions can be made.

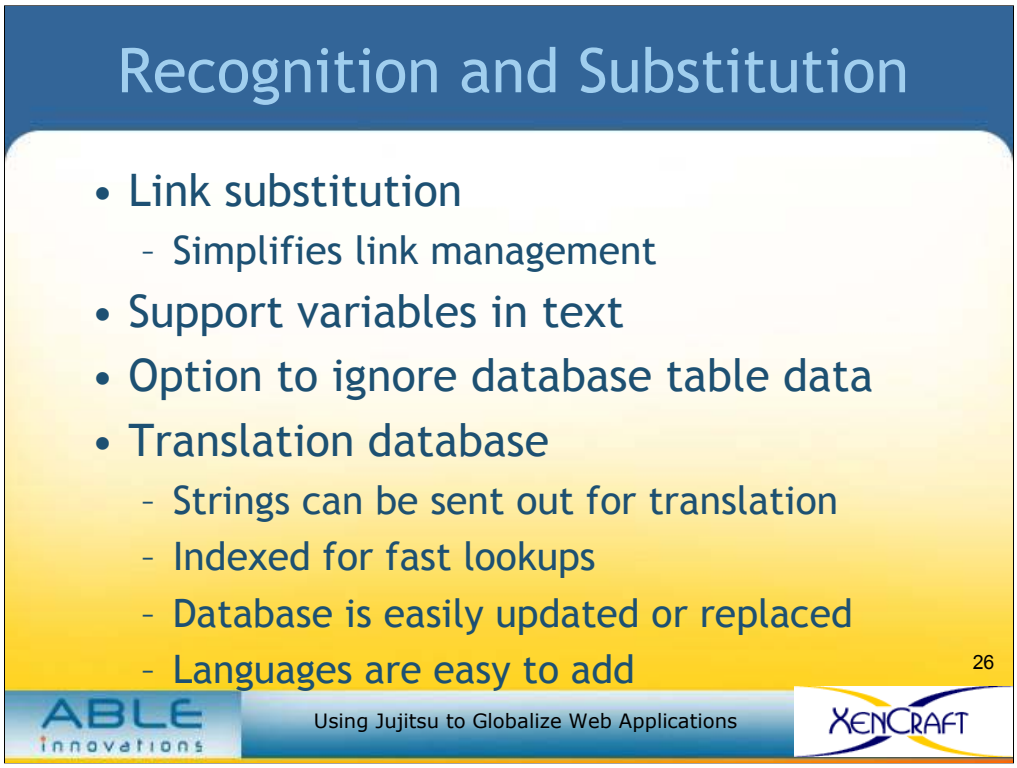


You might think that an HTML page would be parsed into its constituent elements, attributes, and data. However, we have the idea to simply treat the page contents as strings. The database then consists of strings which may contain large multi-element fragments of the web pages and their associated translations. The Localization Proxy Server can then attempt to recognize the longest string that compares identically and substitute the translation for it.

For sections that have no match in the database, the original string can be used.

It is also possible to distinguish strings by their context and usage. As localizers will readily testify, it is often the case that two identical strings require different translations. For example, orange the fruit versus the color, or a word like “print” used as a noun vs. a verb.

A nice feature is to be able to hide data that is translated, so it is easy to identify the text that still requires translation.



Because we are substituting dynamically, it becomes very easy to modify and replace links. For example, it is easy to append a language identifier to a link, so that a page identifier such as “index.html” can become “index_en.html” or “index_fr.html”.

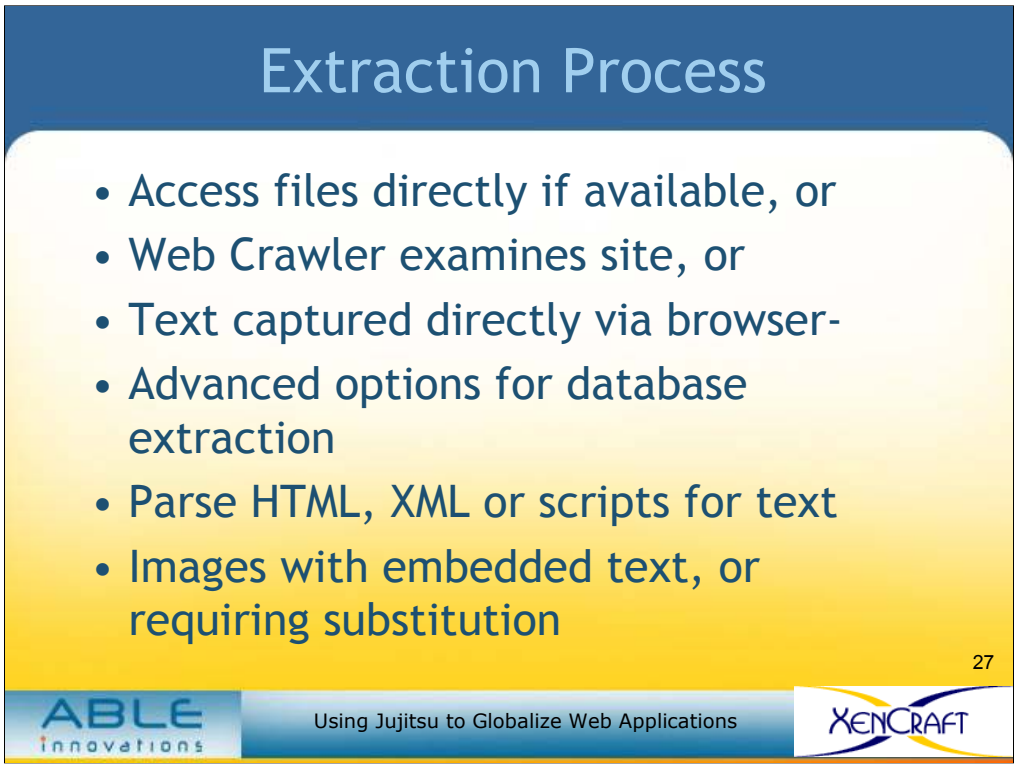
Sometimes data in the web page is variable. It may be programmatically generated (such as today’s date) or it might come from a dynamically changing database in the “web server black box”. It would be difficult and inefficient to populate the translation database with every possible value. So our Localization Proxy Server should support the representation of variables in otherwise fixed text. We can use a common programming syntax such as curly braces around a number.

For example: “You are visitor number 1,234,567!”

Would match: “You are visitor number {1}!”

The matching algorithm would evaluate strings from the original web page to see if they match strings with variables in the translation database, and if so, to perform the appropriate substitution.

Often data from a database is presented in tables and is not to be translated. The Localization server can optionally ignore searching these tables to perform translations.

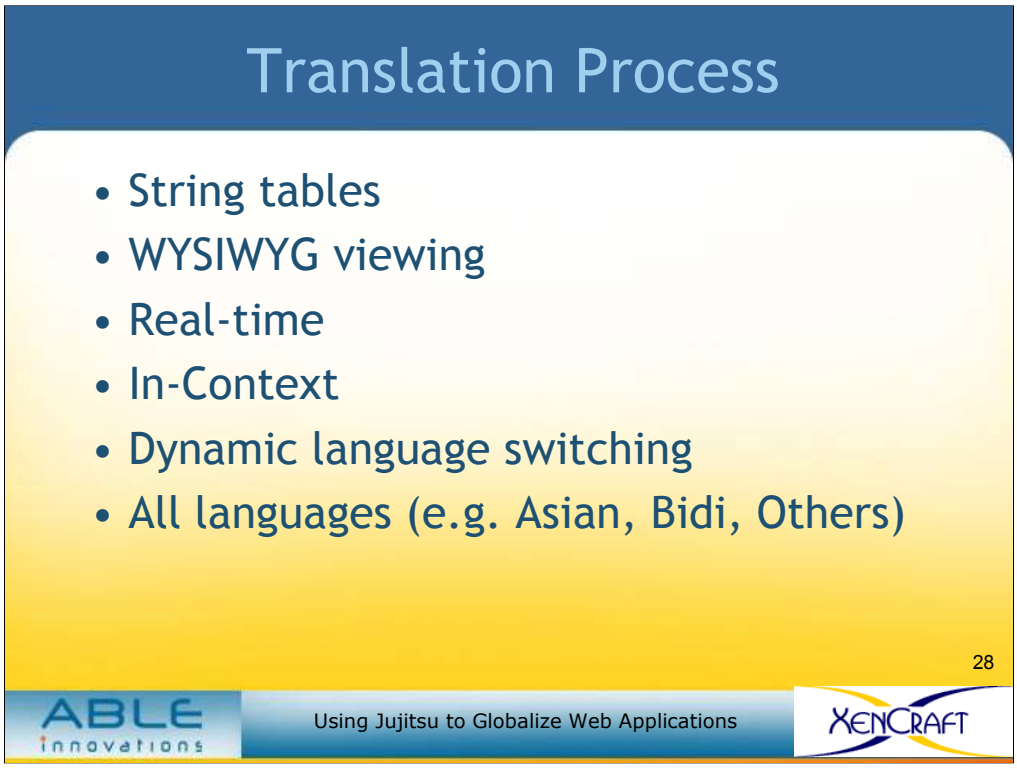


To facilitate populating the translation database, several techniques can be used. The original source files can be accessed and scanned directly.

A web crawler can walk the site.

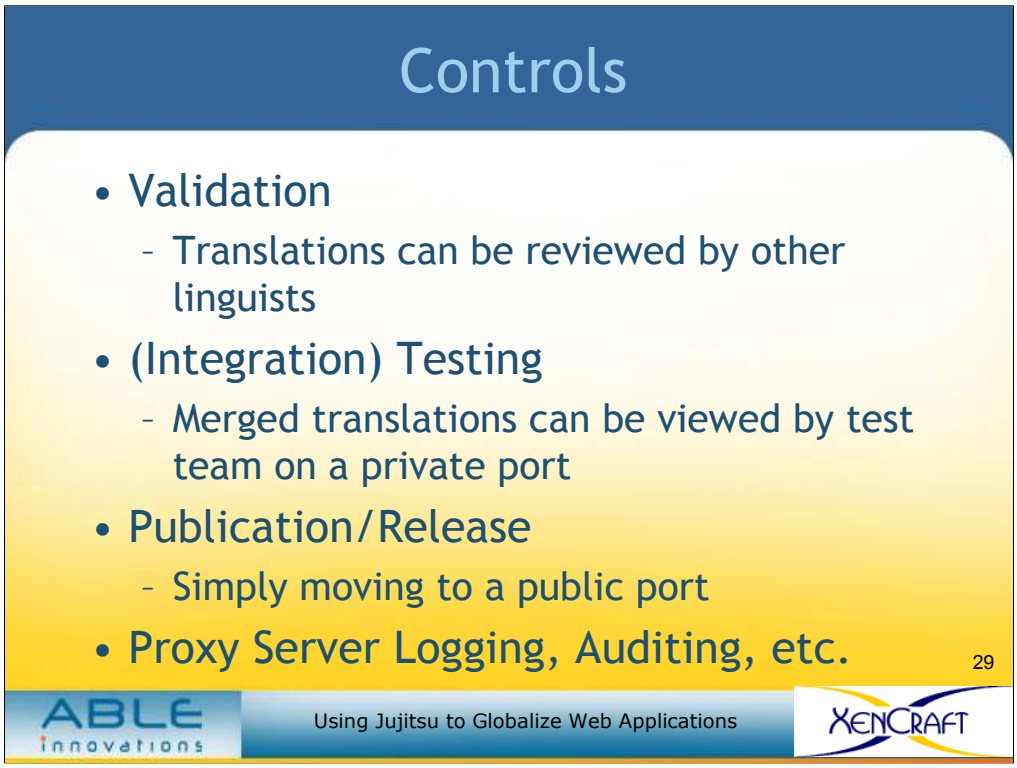
Of course, the pages can be accessed directly via a browser.

And facilities to extract data from a database can also be provided although typically the database contents are dumped and the extraction is made from the dump file.



The translation process is supported by making the string tables available to the localizers. As translations are entered, the resulting pages can be viewed, providing WYSIWYG (What you see is what you get) and in-context viewing.

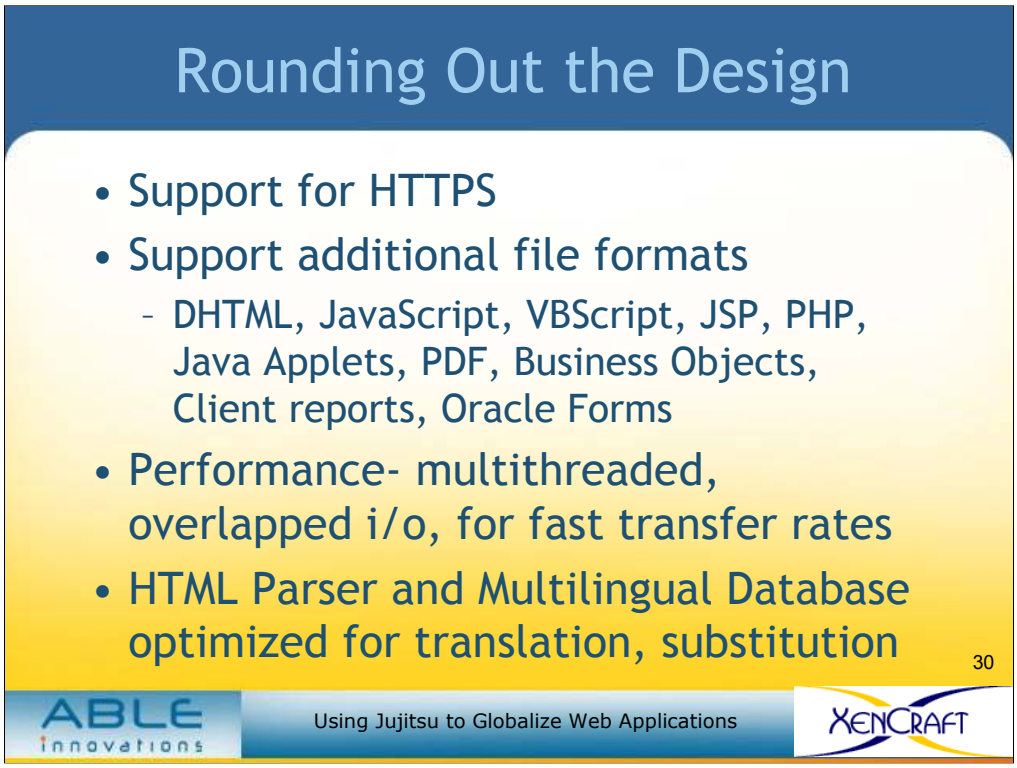
All languages are supported.



As the Localization Proxy Server can make the localization available immediately to a particular user community, (through access controls) the translators work can be reviewed by other linguists as needed.

Similarly, the QA team can access the results and begin performing testing.

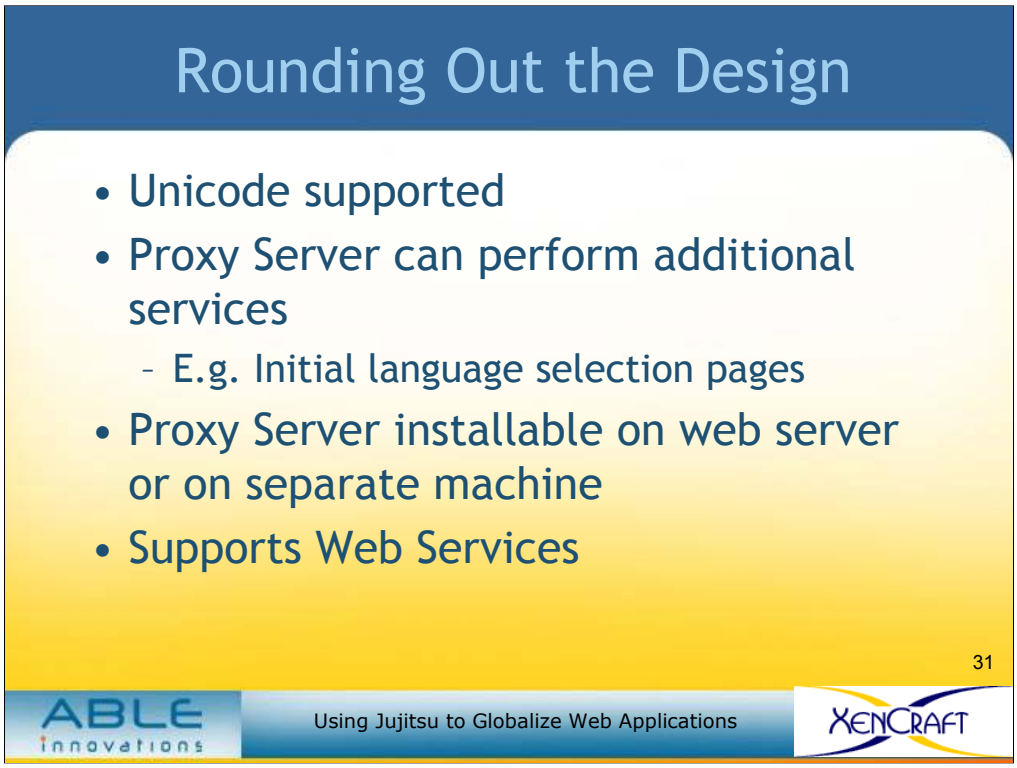
Granting access to the public is a very simple operation.



Here are some other considerations that we can easily take into account.

Security can be managed thru HTTPS support.

Recognition for many file formats and performance optimizations increase the effectiveness of the Localization Proxy Server.



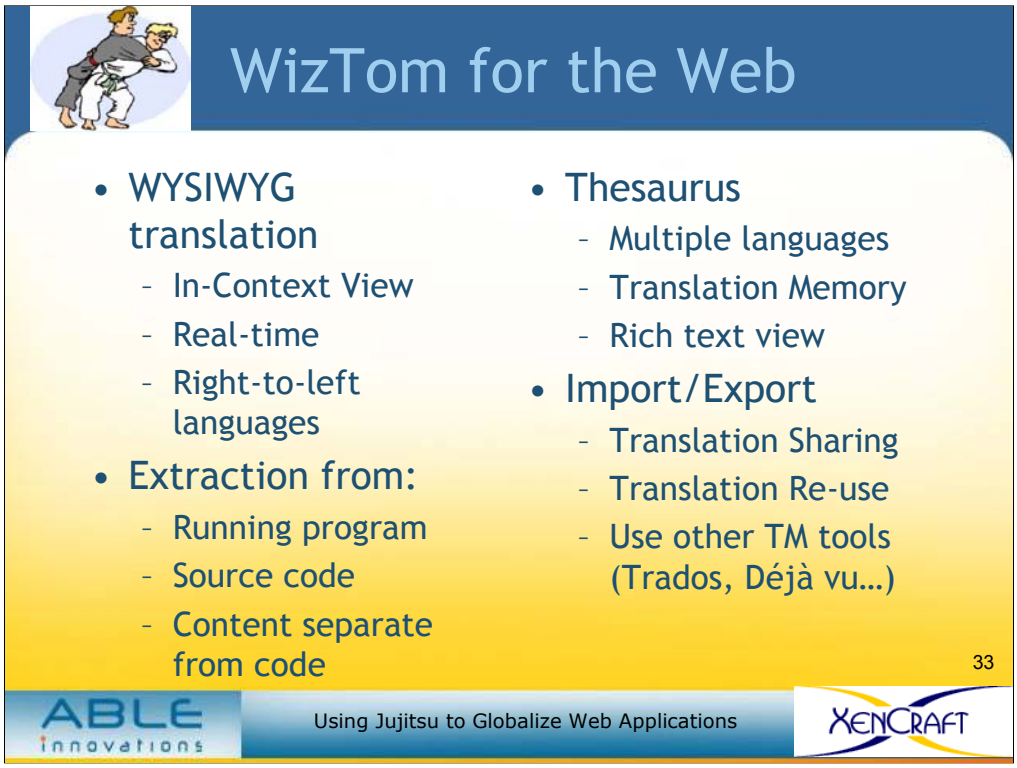
Of course, Unicode is supported.

A nice feature is that the Localization Proxy Server can query the user to select a language, so that this does not need to be implemented in web pages.

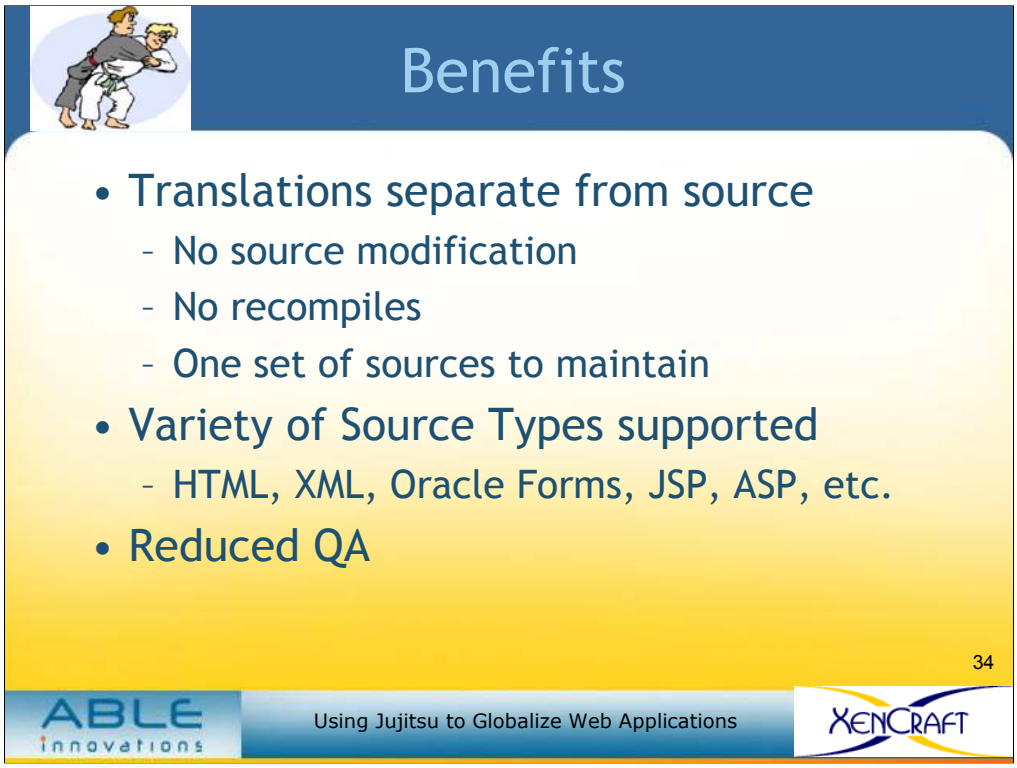
The proxy server can be on the same machine as the web server, or to enhance performance can be installed on a separate machine.

Well, we have implemented our innovative idea and it is now a product available from ABLE Innovations, called WizTom for the Web.

This a screen shot of its user interface.

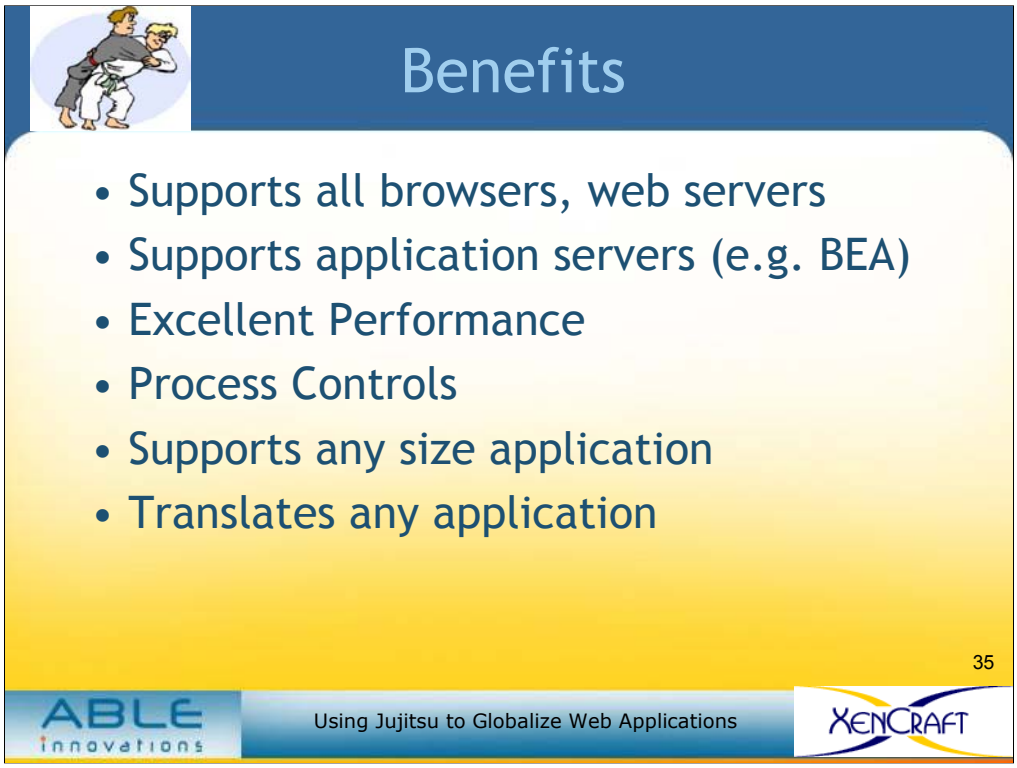


Here are some of the many capabilities of WizTom for the Web



Among the benefits, one of the biggest is the savings in maintenance and QA of the web site. To the extent that the programming of the site is not changed and the translations are substituted in the final resulting pages, there is little need for functional QA of the translated pages. Only a linguistic QA need be performed. This greatly reduces the cost of localizing a site.

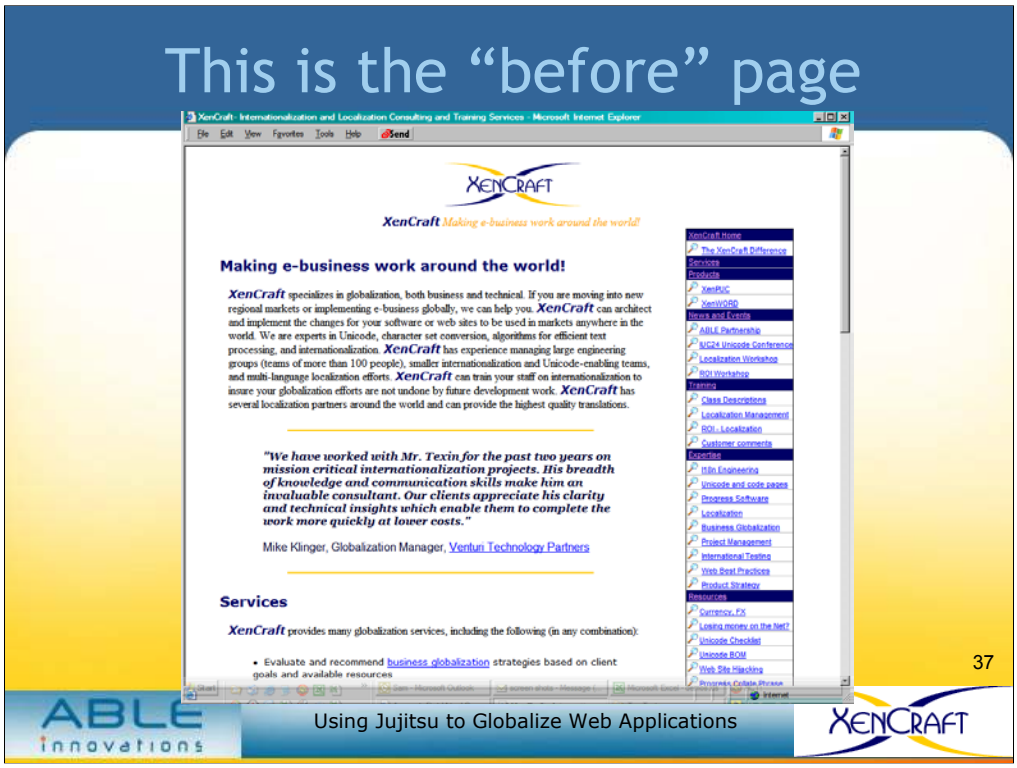
It is also a tremendous savings on site maintenance, since any programming changes only needs to be made once and not multiple times for each language.



This approach is not limited by browser or server versions or brands, and it is scalable for any size site or application.

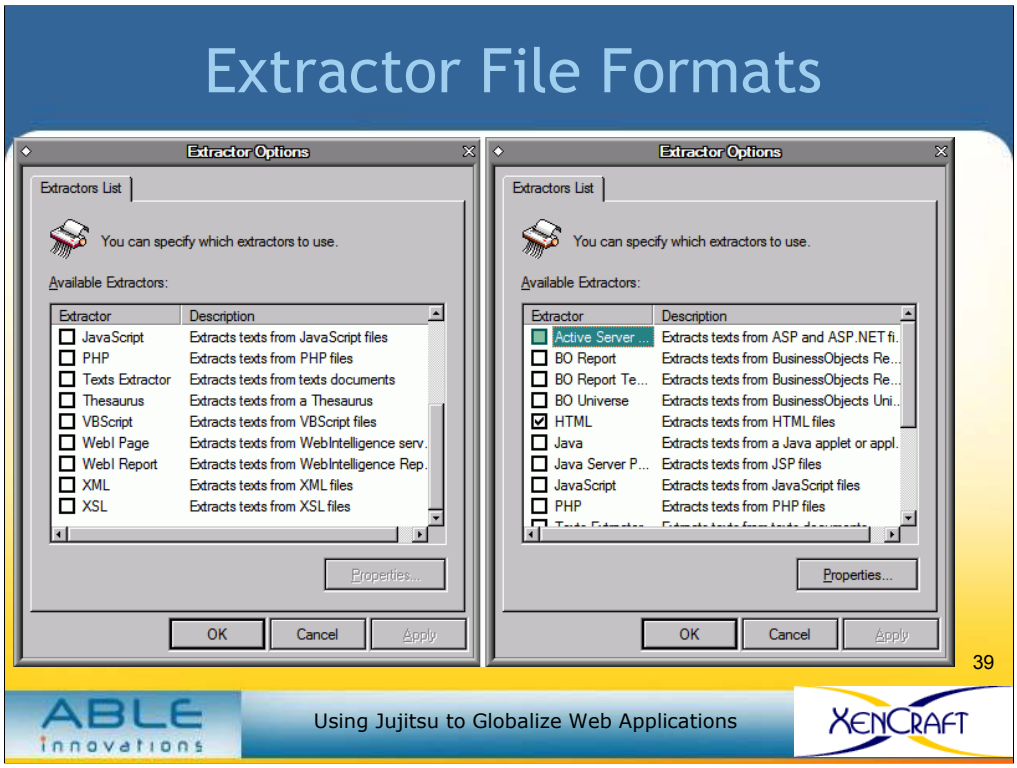
To demonstrate how effective WizTom for the Web is, we will use it to translate one of the web pages from Tex's XenCraft site into Japanese. The screen shot is showing what a translator would use and see. WizTom for the Web is showing the table of strings to be translated, and their Japanese translations in the middle 2 panels. The bottom 2 panels show the current string being translated. The top panel is giving a WYSIWYG view of the translated page.

The right side panels are menu command options.



This is the English page that will be translated.

Here is the Japanese version that Bill created.



In addition to HTML, WizTom for the Web can be used to translate other file formats. Here are the formats that it has “extractors” for, and so can directly read and parse these files to extract strings into its translation database.

After the conference, questions can be sent to either:

Tex Texin tex@XenCraft.com or

Bill Kirtz bkirtz@ableinnovations.com