Designing for International Users: Practical Tips

Richard Ishida
W3C Internationalization Activity Lead

Online version available from
http://www.w3.org/2007/Talks/0706-atmedia/
Terminology

**Localization**

The *adaptation* of a product, application or document content to meet the language, cultural and other requirements of a specific target market.

**Internationalization**

The *design and development* of a product, application or document content that *enables* easy localization for target audiences that vary in culture, region, or language.
Linguistic issues

- Separating content & presentation
- Cultural issues
- What this means to me
In this first section we will look at a few ways in which languages differ, and then see those differences causing practical issues for localization where the developer/designer has not thought about internationalization.
This shows four different ways of writing one idea. In each case the order of 'words' and the number of 'words' is different.
This slide shows how the English word 'On' can map to three different words in Spanish. And then there are the masculine, feminine and plural forms of agreement that change the shape of the word according to its context.
In Russian there is a complex plural system. Apart from the irregular teens, the word endings are applied in a rotating way.

<table>
<thead>
<tr>
<th>Number of pages</th>
<th>Russian word for ‘page’</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>страница</td>
</tr>
<tr>
<td>2-4</td>
<td>страницы</td>
</tr>
<tr>
<td>5-10</td>
<td>страниц</td>
</tr>
<tr>
<td>11-20 (irregular)</td>
<td>страниц</td>
</tr>
<tr>
<td>21</td>
<td>страница</td>
</tr>
<tr>
<td>22-24</td>
<td>страницы</td>
</tr>
<tr>
<td>25-30</td>
<td>страниц</td>
</tr>
<tr>
<td>&gt; 30</td>
<td>Repeat pattern of endings for 1 to 10.</td>
</tr>
</tbody>
</table>
This slide introduces the idea that terms or labels can be of widely differing lengths in different languages.
In languages such as German, Dutch or Swedish it is common to find English ‘compound nouns’ expressed as a single, long word.
Composite messages
For this slide we imagine that the W3C Validator is altered slightly so that it tells you how many validation errors are in your file. It will do this using a ‘composite message’ whose parts are assembled using PHP code as the page is served.

Although we use PHP for these examples, the concepts can be applied to other scripting or coding environments.
In the German translation, the order of the two variables may need to be changed.
Typically translators have no access to the actual code, to avoid them introducing bugs into the page. Either the text is extracted or a translation tool masks the code.

Although we are fortunate that we were able to add words after the second variable, due to the English string containing a period, this still didn't produce the right result. The German reads "File 268 contains myFirst.html validation errors."
The reason is that the translation process didn't switch the order of the variables.
So next we try using a printf statement. This has the benefit that text and variable locators all sit within a single string, and the translator can access the items they want to reorder.

Unfortunately, this doesn't help, since PHP still replaces the variables in the string in the order of the variables cited in the following parameters to printf. This causes the 268 to be shown instead of the filename, by converting the integer value to a string. It is unable to find an integer value in the file name, and so presents us with the zero for the number of errors.
By embedding the variable names directly in the printf string, as shown in this slide, we finally achieve the desired result in German.

Nota bene: Successful, or at the very least, cost effective localization in this case is down to the designer/developer understanding the potential pitfalls of various approaches to coding. It is not the job of the localization vendor to get this right. It needs to be done as the initial content is created!

You should also be very careful of the assumption that "This doesn't affect me, since we don't translate the content I develop." I have seen many, many cases where the thing being developed was later so successful that people wanted to take it to other regions, only to find that they ran into major difficulties because of issues with the translatability of the code or content. It's best to just do it right from the start.
By the way, there is a way to produce the right effect while using the %d and %s variable markers in a PHP string, but it involves a slightly more complex syntax. This is shown in the above slide. The numeric markers refer to the relevant variable in the parameters that follow the string, even after reordering.
So now we know how to code this type of text in PHP... or do we?

Let's think back to our example of how plurality works in Russian, and we realize that we still have a problem for that language. We only have a single string and it can only be translated one way – yet the Russian requires three variants of the word ошибка, depending on the number that precedes it.
To deal with this, the Russian translator would probably resort to a completely different structure for the text, essentially equivalent to "File: X. Validation errors: Y". This approach requires only one form of ошибка in the invariable string. This is an example of what I call a 'topic-comment' composite message.
So we are beginning to see here that there are two distinct types of composite message. The first is based on a sentence-like approach, and the invariant string can be difficult to translate in some circumstances because of the need for agreement or different word mappings.

In the example above, 'The' should be translated 'el', 'la', or 'las' in Spanish, depending on what word follows it. Also the word 'on' should be translated using three different Spanish words (with different endings).
The other approach to designing composite messages is what I like to call the 'topic-comment' approach: you state a topic, then you say something about it.

This approach works much better for the previous slide, since you each comment you associate with a topic can use a different word with the appropriate word endings.

There is a little more to this theory of composite messages than we have mentioned so far, but you can get more information from the W3C Internationalization site at the following URI: http://www.w3.org/International/articles/composite-messages/.
I should, however, mention just one other point. Many designers/developers looking at the English topic-comment arrangement on the previous slide might think to themselves that they could save a little bandwidth by reducing all those instances of the word 'On' to a single string that is used for all comments, ie. they want to re-use strings.
Tempting as this idea may appear, it will unfortunately introduce insurmountable problems for translation, since the comment is likely to require different agreement forms at the least, and possibly different words altogether, depending on the context.

This slide shows an example of how such a problem may come about by returning the same text from a function for each comment.

Note that I do not want to rule out string re-use altogether – there are situations where it is a sensible approach. But re-use must not occur across different contexts. For more information about this, see the W3C Internationalization article at http://www.w3.org/International/articles/text-reuse/.
Now we switch to a very different topic area, that has more to do with the visual layout of the page than the composition of the text.
Let's assume that we want to implement a **fixed-width** box on our page. The text can expand downwards, but not sideways.

Let's also assume that we want a background with a nice gradient behind the title of the box, and that the background has a line across the bottom.

(This slide in Spanish has the title 'Interface Language', and a list of radio buttons to select a language.)
As our text expands during translation into Malay, the title occupies two lines. Unfortunately the graphic used for the gradient background is only one line deep, and things now begin to look a mess.
A way to approach this issue is to use a graphic that is three or four lines deep behind the title. By attaching the graphic using the CSS background property, only the amount needed to view the title will actually be shown.
To get the line to appear in the right place, we simply create it as the bottom border of the heading.

This example uses a technique (and the exact same code and graphic) described in Dan Cederholm's book, Bulletproof Web Design (although the text is borrowed from Google's language preferences). This is significant! Dan is not writing about internationalization per se – he is more concerned with people pumping up the text size for accessibility reasons. It just so happens, however, that the same approach helps with localizability.

This is an example that you don't necessarily have to learn new information to deal with internationalization issues – just following existing best practices can be the key in many cases. Note again, however, that we are still talking about the design and development of content – not about work that the localizers will do!

Dan's book contains several other recommendations that will benefit internationalization.
Note, in passing, an issue related to the Google text I used in the previous example. The dialogue allowed you to select a different language for the user interface from a pull-down list, presumably assuming that your reason for changing was that you couldn't read the current language.

The issue for me is that the names of all the languages are in the language of the current page. Let's assume, for example, that a curious person wanted to see what the interface looked like in Persian, so they selected that language from the list and clicked on the 'Save Preferences' button.
Assuming that they would be able to find their way back to the appropriate dialogue box to get back to English (which would require them to remember which link to hit on the thankfully uncluttered Persian Google home page), that they can remember which is the required select list, and that they can do so in spite of the mirror-imaging of the page when using Arabic script, they would then be faced with what you see on the next slide.
Note that the names of languages are all in Persian, and are sorted by Persian rules.

Which selection would get you back to English ??

(Hint: if you want to explore like this, use a different tab or window for your explorations, and leave the original dialogue available in another for when you want to reset to your current language.)

Of course, the point is really that a Persian person taken to the English site may have as much trouble finding their way to the appropriate user interface language as the curious explorer does in getting back. In my opinion it would help a great deal to write each language name in its own script and language. You can read more about that in the W3C Internationalization article at http://www.w3.org/International/questions/qa-navigation-select.
Let's take a moment to explore another potential issue related to the length of text in translation.

Let's continue to assume a situation where text appears in a fixed width box. We will apply the same approach we discussed earlier to deal with the title of the box. The issue this time will be that we have used a table to apply form labels to the left side of the form entry field to which they apply. Our initial source text is in English.
The English looks nice enough. They Malay, on the other hand, looks pretty ugly. The large expansion factor produces unfortunate stacking of the text on the left, and large white spaces to the right. Although the box expands vertically to hold all the text, we are wasting a lot of space and decreasing the amount of information that will appear in the reader's initial screen (you can imagine that this would be compounded by other fixed with boxes on the page).

With the German translation we have a different problem. The long word Benutzeroberfläche doesn't wrap, and so pushes the select boxes beyond the width of the fixed box container. This has the potential to badly affect the layout of other parts of the screen.
You may want to consider avoiding table cells in such constrained circumstances. This slide shows how the text would look if the input fields were just in a paragraph with the label text.

All the boxes now look fine, and although there is a very slight increase in vertical height overall, we have removed the problems seen with the Malay and German text on the previous slide.

Let's note, again, that this is down to the way the page is designed/developed, not the way it is localized. That's a fundamental message of this presentation. Internationalization during design and development removes significant barriers to deploying your content globally.
Now we are going to look at the benefits to localization of another good design/development best practice that you would hopefully adopt anyway: the separation of content, presentation and behaviour.
People at this conference should be familiar with the idea that content and presentation should be kept separate, if you want manageable and easily maintainable web sites.

Each of these windows shows EXACTLY the same HTML file. The changes made to the CSS file produced three very different presentations of that basic content.

This is particularly useful for changing the presentational aspects of a site or group of pages. You typically only need to edit a single CSS file, rather than editing all the code of each HTML file.

This can also be beneficial for localization, since typographic approaches, colors, etc, may need to be changed for different locales. Making such changes in the CSS is much easier than adapting the HTML.
Here are some ways in which typographic differences may appear between language versions of the same content. It is much easier to apply each of these typographic differences if you can do so via a CSS style sheet, rather than searching through the HTML or script code.
Leading the Web to its full potential...

ปัญหาการเรียงลำดับของข้อความบนหน้าจอ

- problems of resolution to support bold and italics in small characters on-screen => different ways of emphasizing text
- different fonts, and possibly font-size, need to be used for font-family assignments
- line heights, paragraph spacing may need to be adapted, since characters may be more complicated
- problems of resolution to support bold and italics in small characters on-screen => different ways of emphasizing text
- different fonts, and possibly font-size, need to be used for font-family assignments
- line heights, paragraph spacing may need to be adapted, since characters may be more complicated
- wrapping rules may need adjustment (e.g., there are no spaces between words in Chinese)
- no upper- vs. lower-case distinction
- no convention of distinguishing between proportional and mono-spaced fonts
You should also consider separation of content and presentation when adding scripting. Let’s suppose that we wanted to load some JavaScript after this basic test page has loaded which would automatically add a list of tests on the page to the top right corner. (We may actually want to add links to these tests, but I have resisted that temptation so that the following slides will contain the code examples.)
function makeList () {
    var headings = document.getElementsByTagName('h2');
    var div = document.createElement('div');
    div.style.cssFloat = 'right';
    div.style.styleFloat = 'right';
    div.style.width = '20%';
    div.style.padding = '1em';
    div.style.border = '1px solid teal';
    div.style.backgroundColor = '#FFCC99';
    div.style.fontSize = '90%';
    div.style.margin = '1em';
    div.style.color = '#496633';
    var title = div.appendChild(document.createElement('div'));
    title.style.fontWeight = 'bold';
    title.appendChild(document.createTextNode('Tests on this page.'));
    for (i=0;i<headings.length;i++) {
        var test = div.appendChild(document.createElement('p'));
        test.appendChild(document.createTextNode(headings[i].childNodes[0].data));
    }
    body = document.getElementsByTagName('body')[0];
    body.insertBefore(div, body.firstChild);
}
Note how we are adding style information directly to the DOM while running this script. This is really obvious in this example, since there is such a lot of it. It is particularly tempting to do this sort of thing if you just want to add a single style effect, such as bolding, to text.
This version of the same function shows a much better approach. We assign an id attribute to the box, then move all the styling information to a CSS file, referencing the markup via the id. This makes the code much cleaner and makes it easier to manage the styling.

Again, this technique is recommended as a standard best practice in Jeremy Keith’s book Dom Scripting (which contains many other useful ideas along similar lines). It is another good example of how good web design benefits localization.
This section will look at a very different set of issues – those related to cultural differences in design.
International Web Sites

Formatting & reading data
The date at the top of the slide is ambiguous in three ways. This is a bad way to generate dates for a page – it is better to use a name for the month (or, in some specialized cases, you may be able to use a four-digit year and the order year, month then day).

Note also how the expected separators, leading zeros, etc vary from culture to culture.

The biggest issue, however, is not displaying the date correctly, but recognizing a date supplied by a user if you haven’t adequately signposted the order or format in which you expect to receive data. You need to make sure that your expectations are clear if you want to stand any chance of recognizing what the user is typing in.
Often, using a graphical calendar can provide a more user friendly and reliable method for users to indicate dates. (Be careful to ensure that your calendar allows enough space for translations of the month and day abbreviations in other languages, of course!)

Bear in mind, also, that people in some parts of the world use local calendars.
When Chinese people write their name they normally write the family name first and given name last. A form that asks a Chinese user to enter their first and last names can be very confusing for them. Better say ‘family’ and ‘given’ name.

The Malay person above has only one name, Isa. ‘bin’ means ‘son of’, and Aman is his father’s name.

A similar situation applies to the person from southern India whose name appears at the bottom.

There are other ways in which names can vary, including double family names for Spanish people, and patronyms for Russians.

When creating forms for names, ask yourself what you will do with the name. If you won’t process it at all, allow people to enter their whole name as they would usually write it. However, if you are expecting, for example, to use part of the name to address people, you may find that you can’t simply work out what to call people working from Anglo-Saxon expectations of how names are used. You may need a special field that asks how the user likes to be addressed. Also be careful about choosing a part of the name for sorting – people sort names in very different ways around the world.

<table>
<thead>
<tr>
<th>Chinese</th>
<th>Chang Wen Yun</th>
</tr>
</thead>
<tbody>
<tr>
<td>Malay</td>
<td>Isa (bin) Aman</td>
</tr>
<tr>
<td>Indian</td>
<td>M. Thiruselvan</td>
</tr>
<tr>
<td></td>
<td>Thiruselvan s/o Manickavasagam</td>
</tr>
</tbody>
</table>
You also need to consider that addresses look quite different from country to country. Russian and Japanese addresses are written from the general to the specific, top to bottom. You may need to figure out how to produce these different orderings for forms.

Also, the name of the Russian person above is in the dative case (expressing the idea of 'to the person'). If you asked her to simply supply her name in a separate box, she would probably write Юлия Селиванова, rather than Юлии Селивановой. How will you deal with that?
There are a number of ways in which formats differ around the world. Note that recognizing information input into general forms can be more difficult than producing form templates in the right way.
Symbolism & examples
This check symbol means 'correct' or 'ok' in many countries. In some countries, however, such as Japan, it can indicate 'incorrect'. Japanese often convert check marks to circles (their symbol for 'correct') as part of the localization process.
The circles in the columns of this board indicate that space is available, not that there are 0 seats left. It is the equivalent of the check mark.
This illustration of sports items is not representative of sports played in the UK, and may need to be changed.
Gestures and sometimes body language can often give completely the wrong message, and should be used with extreme care.
Color
This phone is likely to be perceived immediately as a public telephone in Japan, due to the conventional use of the green color there. In most other parts of the world, this cue is missing. So colors have conventional roles that differ from culture to culture.
Color names also differ from culture to culture, dependent on context. British people often call the middle light here amber, whereas Americans call it yellow. Japanese speaking English will often refer to the bottom light as blue.
Differing approaches
People do things in different ways in different parts of the world. For example, Lotus 1-2-3 was relaunched in Japan with the radar chart after it was discovered that this was a very common way of representing comparative data there.
In the Middle East, you may find that tables, spreadsheets, collated pictures and the like need to flow right to left, rather than left to right.

Some graphics with directional bias may need to be mirrored or changed for a predominantly right-to-left context.
Then there are more fundamental issues about whether the application, product or solution you are developing will actually fit into the foreign culture at all.

"... one Latin American teacher recently complained to me that the US-manufactured and well-translated educational software currently being used in his country's primary schools presupposed 'solitary problem solvers', whereas his culture stressed collective problem-solving."

The following slides show how Yahoo **adapts the content** on its various local home pages, rather than just translating it. This may be something you also need to consider.
International Web Sites
### Cultural issues

#### Differing approaches

<table>
<thead>
<tr>
<th>Category</th>
<th>Subcategories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yahoo! Japan</td>
<td></td>
</tr>
<tr>
<td>Entertainment</td>
<td>メディアお楽しみ、テクノロジ、テレビ、ラジオ、新聞、雑誌...</td>
</tr>
<tr>
<td>Sports and movies</td>
<td>ビデオお楽しみ、映画、ポップ、ロック、ヒップホップ...</td>
</tr>
<tr>
<td>Music and art</td>
<td>番組、編集、美術、歴史、文学...</td>
</tr>
<tr>
<td>Life and culture</td>
<td>コミュニティ、インターネット、ハードウェア、ソフトウェア、WWW...</td>
</tr>
<tr>
<td>Education</td>
<td>教育、大学、専門学校、小中高、幼稚園...</td>
</tr>
<tr>
<td>Health and medicine</td>
<td>輔助、健康、ライフスタイル...</td>
</tr>
<tr>
<td>Social science</td>
<td>建築学、社会学、言語、政治学...</td>
</tr>
</tbody>
</table>

(Copyright © 2006 W3C (MIT, ERCIM, INRIA))
Even if you expect readers to use English on your site, be careful how you write your English. Don’t expect people who use English as a second language to understand all the idioms, words or concepts you are familiar with. (This slide is just an example to make the point that English people may not even understand American texts – see the bits in blue.)

Also consider the complexity of your grammar. Short, simple sentences can often help non-native speakers.
Montre qui tu es!

The 5 R's

Nothing sucks like an Electrolux!

Of course, look out for difficult situations when it comes to translation.
Be wary also of visual puns – i.e. graphics that rely on the user speaking a language to be understood.
For example, the French and Japanese translations of ‘Hang up’ have nothing to do with ropes or things hanging. If you translate the icon label, a French or Japanese person will most likely be confused by the choice of graphic!
If you wanted to translate this text to Russian, and you were supplied with a jpeg file, you’d have to carefully rub out the English, then redraw the complicated background, before you could finally add the Russian text. That would take a huge amount of time. Alternatively, if you were provided with a layered file, that puts text on one layer and background on another, it would be very quick and easy to produce the translation. Think, therefore, about how you go about the process of handing things off for localization.

Of course, an even better approach would be to use CSS positioning for the text. That would make it searchable and selectable. Try to avoid using readable pixellation when you can.
What this means to me

Linguistic issues
Separating content & presentation
Cultural issues
What this means to me
This slide summarizes some of the practical takeaways from this presentation.

The presentation is not designed to give you a thorough overview of potential internationalization and localization issues – we would need longer for that. It aims to provide you with a few practical takeaways, but more importantly it aims to get you thinking about what internationalization is all about – to take you out of your comfort zone, and help you realize that if you want your content to wow people outside your own culture and language, you need to build in certain flexibilities and adopt certain approaches during the design and development – not as an afterthought. Otherwise you are likely to be creating substantial barriers for worldwide use.

The presentation also aims to show that, although using Unicode is an extremely good start to making your stuff world-ready, using a Unicode encoding such as UTF-8 throughout your content, scripts and databases is only a start. You need to worry about whether translators will be able to adapt your stuff linguistically, but you also need to also consider whether graphics and design are going to be culturally appropriate or can be adapted, and whether your approaches and methodologies fit with those of your target users.
International Web Sites

What this means to me
Internationalization means:

- using a Quality approach to reduce the overall cost and time to market/release of multinational deliverables
- designing into the deliverable or product an internationalized base, and a modular and easily adaptable architecture – don’t treat internationalization as an afterthought!
- not always doing extra work – maybe just working in a better way
Remember also that, even though you think you don’t deal with content that will be internationalized now, you may well need to in the future.
The W3C is trying to provide useful advice at http://www.w3.org/International/

We could always do with help and support for this.
Remember:
Portability, scalability, interoperability + localizability!

Thank you
http://www.w3.org/International/