

Internationalisierung

- wenn der Webauftritt mehrsprachig werden soll wird es kritisch

Vorab eine Definition und Abgrenzung

Den Prozess, eine Anwendung so zu entwickeln, dass sie sich automatisch an die verschiedenen Regionen und Länder anpasst, ohne die Anwendung nochmals neu kompilieren bzw. Code schreiben zu müssen, nennt man Internationalisierung (engl. Internationalization).

Dieses Wort Internationalization enthält 18 Buchstaben zwischen dem beginnenden I und dem endenden n, so dass man häufig auch die Abkürzung I18n benutzt.

Da es aber weniger um die internationale Kommunikation geht als um die Anpassung der Anwendung und GUI an eine Sprache ist der Begriff "Localization" die treffendere Bezeichnung.

Es gibt natürlich noch ein weiteres Themenfeld: Über die Internationalisierung und Globalisierung der Wirtschaft möchten wir uns aber hier nicht auslassen.

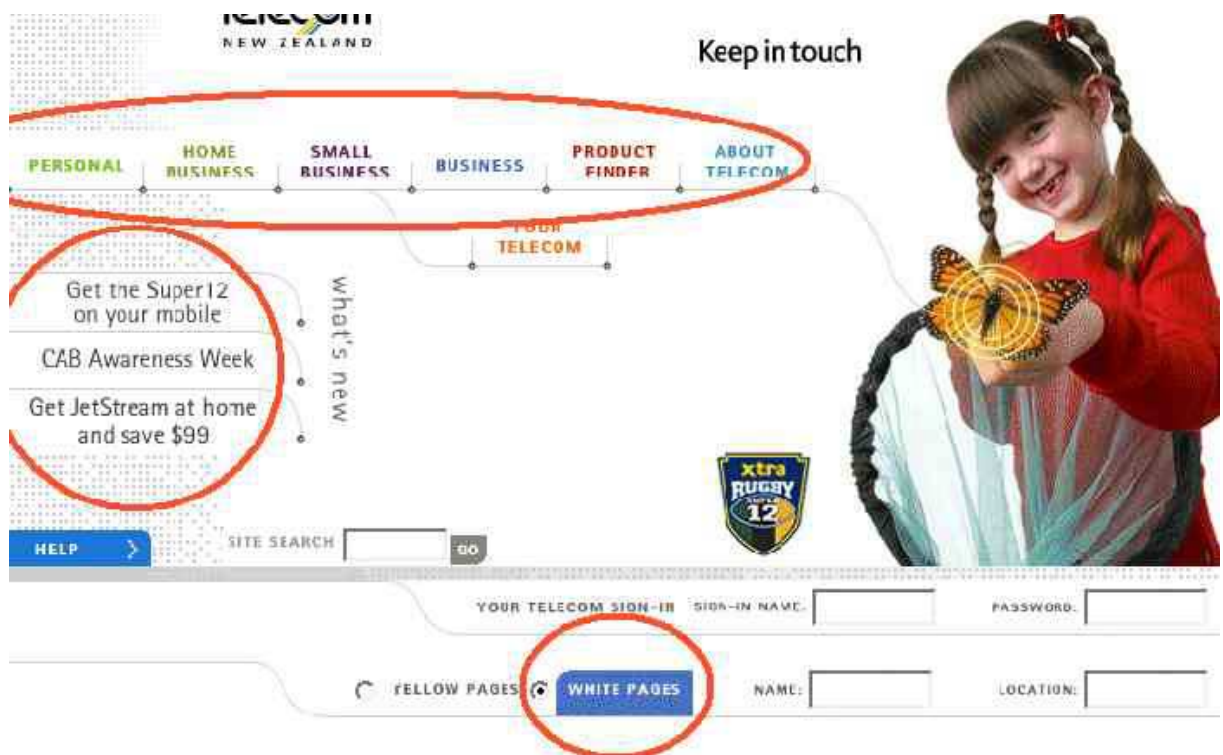
Mehrsprachenfähigkeit ist eine wesentliche Anforderung an heutige Anwendungen. Umso erstaunlicher ist die Tatsache, dass gerade im Bereich der Internetauftritte und Portale durchgehend die notwendigen Regeln ignoriert werden. Man könnte sogar so weit gehen anzunehmen, an diesen Internetpräsenzen hätten sich Vorschulkinder ausgetobt – und natürlich die Grafiker.

Was ist damit gemeint?

Navigationselemente sind in überwiegender Manier als Grafikobjekte realisiert. Das, so wird häufig suggeriert, würde „besser“ aussehen und ist in der Version 0.1 des Internetauftritts auch einfacher zu realisieren. In dem zufällig ausgewählten Screenshot markieren die roten Kreise diese grafischen Navigationselemente.

Dazu kommt, dass die eigentlichen Inhalte in Form von Texten direkt im HTML Code der statischen Seiten zusammen mit dem Layout vermengt werden.

So eine Website erfüllt nur minimal die Anforderungen an einen nationalen Internetauftritt, wie er typischer Weise in der 1. Generation umgesetzt wurde.



Auch wenn folgende Ausführungen generell für alle Anwendungen gelten, soll konkret an dem folgernden Beispiel-Webauftritt diese Idee verdeutlicht werden.

Beispiel: Multilingual Website

Historisch gesehen sind die meisten Webauftritte als statische, in einer Sprache realisierte Websites entstanden.

Konzeptionelle und technische Schwachstellen der konventionellen Websites in Bezug auf Anforderungen an einen mehrsprachigen Auftritt erfordern zurzeit hohe Aufwände bei der Erweiterung des Auftritts hinsichtlich einer neuen Sprachvariante. In der Regel entstehen dabei durch Duplizierung, Übersetzung der Texte und Bereitstellung grafikbasierter Navigationselemente parallele Websites und redundante Objekte.

Hinzu kommt, dass die Pflfegbarkeit durch die Spracherweiterung leidet und damit zu rechnen ist, dass Inkonsistenzen und Fehlerrate steigen. Selbst mit einem modernen Content Management System lassen sich parallele, durch Duplizierung entstandene Websites schwer pflegen. In vielen Fällen liegen die Redaktionen bei den nationalen Dependancen, sodass zunehmend Layout und Corporate Design verloren gehen.

An dieser Stelle sollte in einem international agierenden Unternehmen eine genaue Bestandsaufnahme, Analyse und Identifikation der strategischen Ziele der Website erfolgen und ggf. eine Neukonzeption, wie sie hier beschrieben wird, vorgezogen werden.

Zuerst aber einmal - warum wird eine multilinguale Website benötigt

Die Gründe und Motivation kann sehr unterschiedlich sein:

- gesetzlicher Auftrag (Behörde Schweiz)
- Marketing und Vertrieb in konkretem Sprach-Kulturkreis

- Komfort für anderssprachige Zielkunden (Türkisch in Deutschland)
- Alleinstellungsmerkmal in der Branche

Die technischen Ansätze reichen von vollständig unabhängigen Websites, d.h. Inhalt, Navigation und Formatierung werden separat gepflegt – bis durchgängig parallel in dem Sinne, dass Navigation und Layout identisch sind und Texte in der jeweiligen Ausprägung zur Laufzeit bereitgestellt werden.

Technischer Lösungsansatz

Der folgende Abschnitt wird jetzt ein wenig technischer, kann aber ohne Einschränkung des Verständnisses übersprungen werden.

Das vorliegende Konzept basiert auf den Konstrukten der Internationalization unter J2EE und im Besonderen auf folgende Klassen:

- Date
- Calendar
- DateFormat
- Locale
- ResourceBundle
- Collections

Mit Date, Calendar und DateFormat werden landesspezifische Darstellungen von Datum, Uhrzeit, Währung, Dezimalpunkt usw. erfasst.

Jede Locale repräsentiert eine besondere Auswahl des Landes und der Sprache.

Die Klasse `java.util.ResourceBundle` stellt grundlegende Tools zur Verfügung, mit denen Nachrichten in verschiedene Sprachen unterstützt werden.

`PropertyResourceBundle` - eine der Standardimplementationen von `ResourceBundle` erlaubt es, Ressourcen unter Verwendung der Syntax "name = value" zu definieren bzw. sie zu verwenden. Dies ist eine sehr bequeme Möglichkeit, Resource-Bundles mit Nachrichten zu befüllen, die in einer Webanwendung verwendet werden.

Die `ResourceBundle` Klassen werden demnach benutzt, um Ressourcen vom Quellcode zu isolieren. Man erhält so die Möglichkeit beliebig viele `ResourceBundle` Klassen zu implementieren und kann so beliebig viele Lokalitäten sprich Sprachen abbilden und unterstützen. Java lädt dann automatisch abhängig von den Lokale-Einstellungen das richtige `ResourceBundle`, dass für die gewählte Sprache erforderlich ist.

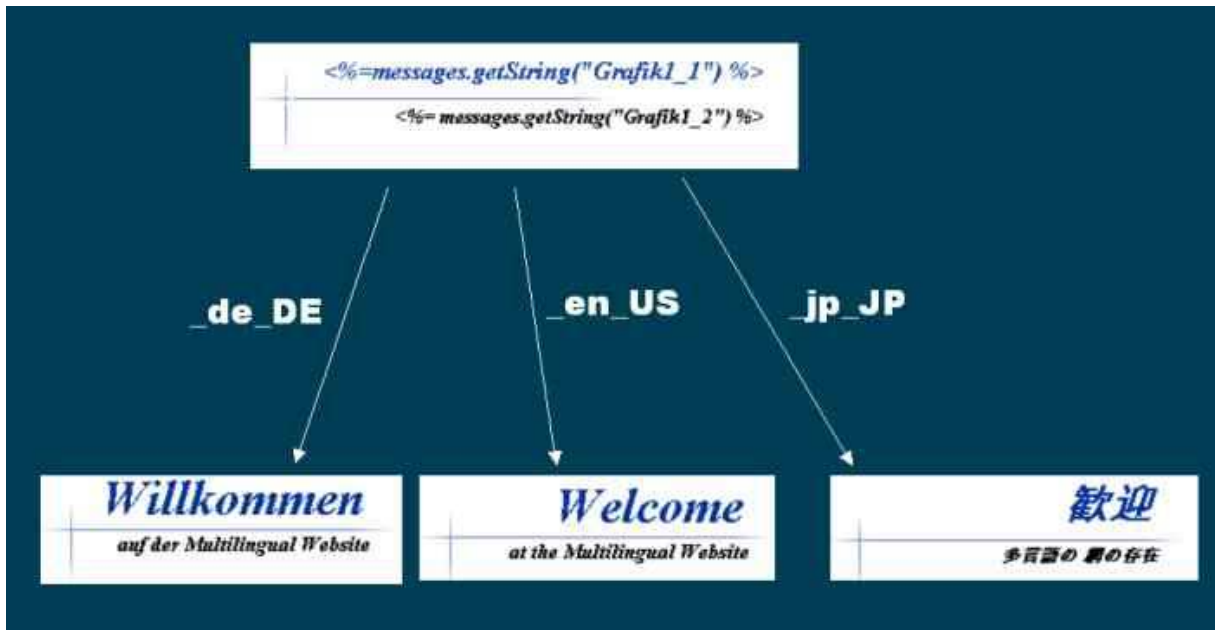
Java benutzt Unicode intern, um seine Zeichenketten darzustellen. Unicode vereinfacht die Zeichenketten Behandlung für DBCS (DoubleByteCharakterSet) Länder, da ein Zeichen immer in zwei Bytes abgelegt ist

Unicode basiert darauf, binär codes zu verwenden um Text oder Zeichen darzustellen. Offiziell wird es der Unicode Worldwide Charcter Standard genannt und ist das System für das Austauschprozess und Anzeigen von geschriebenem Text von unterschiedlichen Sprachen. Momentan enthält der Unicode Standard 34168 einzelne kodierte Zeichen von 24 unterschiedlichen Ländern.

Zum Beispiel werden folgende Zeichen aus dem Japanischen Site als `#25152 #12398 #22320 #22259` abgelegt.

Konzeptioneller Lösungsansatz

Die wesentliche Grundidee besteht in der stringenten Trennung von Text und Darstellung einschließlich Navigation. Grafiken kommen nur insofern zum Einsatz, als sie keine Textelemente beinhalten. Damit das deutlicher wird, hier ein konkretes Beispiel:



Die resultierende Grafik setzt sich aus einer Tabelle zusammen, deren Zellen zum einen Grafiken beinhalten und somit das blaue Kreuz erzeugen und zum anderen aus Platzhaltern, die zur Laufzeit mit den entsprechenden Werten besetzt werden.

Damit die Seiten richtig angezeigt werden können, sind die Quellen nicht mehr in HTML geschrieben sondern als Java Server Pages(JSP) implementiert. Eine Servlet Engine übersetzt dann diese Seite zur Laufzeit und übergibt das Resultat an den Browser.

Genauso kann man auch mit den Navigationselementen verfahren, d.h. sämtliche Links, die konventionell häufig als Grafik ausgeprägt sind, werden wie oben skizziert als HTML bzw. JSP kodiert.

Für jede Sprache wird eine Datei bereitgestellt, die für die verwendeten Platzhalter einen Eintrag bereitstellt. Diese Dateien haben einen sehr einfachen Aufbau:

```

abcdefgh = ABCDEFGH
Grafik1_1 = Willkommen
Grafik1_2 = auf der Multilingualen Website
ug ukjhkj = iououoiuu
...

```

bzw für Englisch:

```

abcdefgh = ABCDEFGH
Grafik1_1 = Welcome
Grafik1_2 = at the Multilingual Website
ug ukjhkj = iououoiuu
...

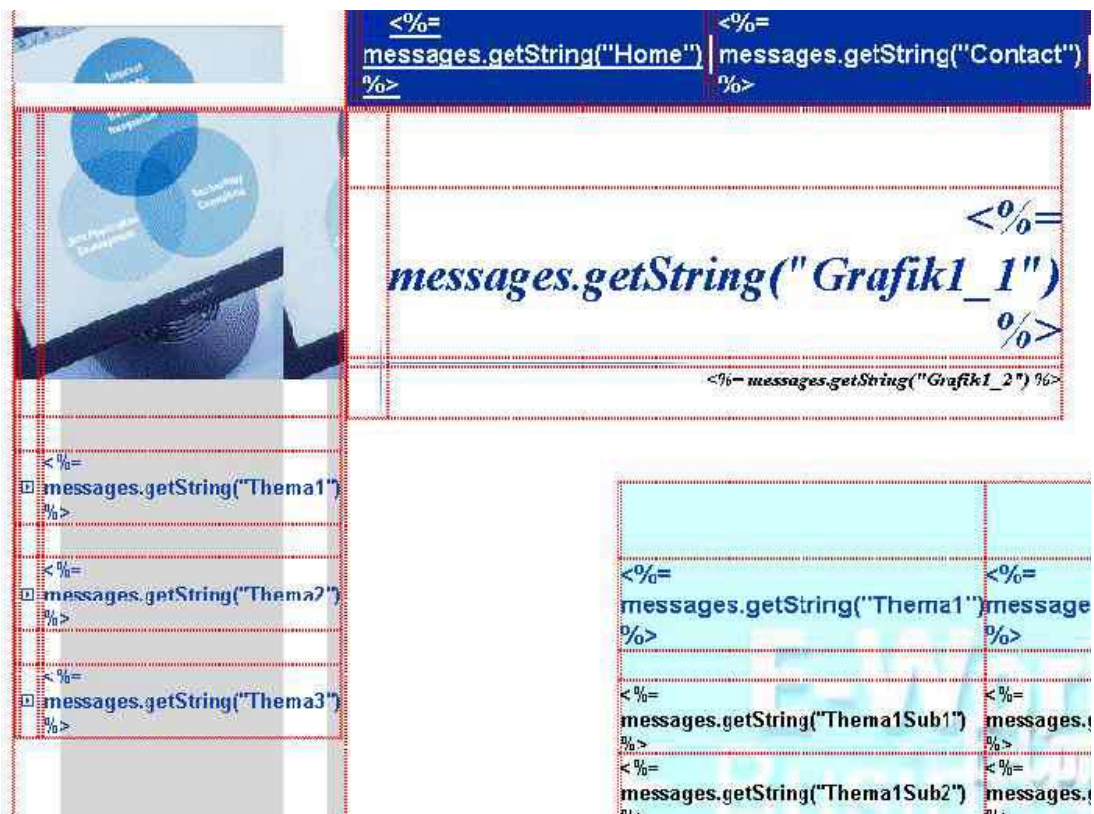
```

Wenn alle textlichen Komponenten auf diese Weise in den JSP Seiten verknüpft sind, besteht die Website aus einem einzigen JSP -Baum. Wenn eine weitere Sprache hinzugefügt werden soll, ändert sich am Programmcode der Website nichts. Einzig eine weitere Datei mit den übersetzten Textbausteinen muss erstellt und der Servlet Engine zur Verfügung gestellt werden.

Demo Site



Das entsprechende JSP "Gerippe" sieht dabei so aus:



Nutzen

Die Architektur des hier skizzierten Konzeptes garantiert eine stabile und leistungsfähige Anwendung für einen multilingualen Webauftritt. Statt mehrerer, paralleler Websites jeweils für eine Sprachausprägung, existiert eine einzige Website.

Reduktion der Komplexität !

Durch die stringente Trennung von Texten und Layout ist diese Website darüber hinaus leichter zu pflegen. Die Vorlagen werden als Templates einmalig implementiert und in einem Content Management System (CMS) verwaltet.

Saubere Trennung von Layout und Texten !

Die sprachspezifischen Dateien lassen sich mit jedem Texteditor lesen bzw. bearbeiten. Soll eine weitere Sprache unterstützt werden, kann der Übersetzer diese Datei verwenden, ohne die eigentlichen Webseiten editieren zu müssen.

Somit drastisch reduzierte Aufwände !

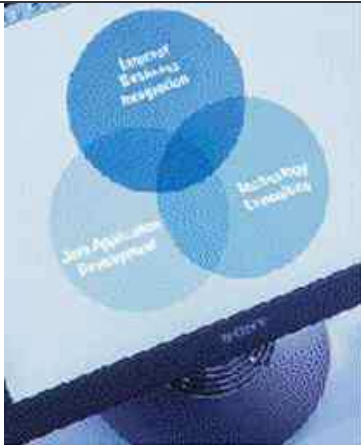
Die Bereitstellung einer Basisversion für alle Sprachvarianten und die Möglichkeit eine Standardsprache zu definieren, so dass bei einer fehlenden Übersetzung automatisch auf die Standardsprachversion geschaltet wird, führt zu einer hohen Konsistenz.

Konsistenz und reduzierte Fehlerquote !

Die Verwaltung der Site erfolgt wie bei einer konventionellen Website mit einem CMS. Alle statischen Komponenten wie Grafiken, Templates, zu inkludierende HTML Dateien und die Sprachdateien selber unterliegen damit dem Pflegezyklus. Dieses Vorgehen hat außerdem den Vorteil, dass eventuelle Restriktionen durch eine Firewall vermieden werden.

Beibehaltung der CMS Infrastruktur !

Und wenn es dann eines Tages auch Japanisch sein soll, freuen sich alle IT Verantwortliche, die diese Konzept umgesetzt haben.



歓迎

多言語の 網の存在

☐ プロダクト

☐ 情報

☐ 会社

プロダクト

情報

相談

組織図

サービス

位置

認識

出版物

さようなら - das heißt Auf Wiedersehen

Haben Sie Fragen zu diesem Thema oder sind Sie interessiert an weitergehenden Informationen, dann rufen Sie uns an. Sie erreichen uns telefonisch unter +49 40 2714340-0 oder per E-Mail: info@evodion.de.