

Gregor Gawol

Lokalisierung von vertikalen Standardapplikationen mit  
Hilfe von XLIFF

BACHELORARBEIT

HOCHSCHULE MITTWEIDA (FH)

---

UNIVERSITY OF APPLIED SCIENCES

Mathematik/Physik/Informatik

Mittweida, Oktober 2009

Gregor Gawol

Lokalisierung von vertikalen Standardapplikationen mit  
Hilfe von XLIFF

eingereicht als

BACHELORARBEIT

an der

HOCHSCHULE MITTWEIDA (FH)

---

UNIVERSITY OF APPLIED SCIENCES

Mathematik/Physik/Informatik

Mittweida, Oktober 2009

Erstprüfer: Prof. Dr. Mario J. Geißler  
Zweitprüfer: Dipl. Kfm. Falk Jahr

vorgelegte Arbeit wurde verteidigt am: 24.11.2009

## **Bibliographische Beschreibung**

Gawol, Gregor:

Lokalisierung von vertikalen Standardapplikationen mit Hilfe von XLIFF.  
- 2009. - 48 S.

Mittweida, Hochschule Mittweida (FH) - University of Applied Sciences,  
Fachbereich Mathematik/Physik/Informatik, Bachelorarbeit, 2009

## **Referat**

Angesichts der Globalisierung ist es wichtig, dass ein Softwareunternehmen seine Software an den internationalen Markt anpasst.

Ziel der Bachelorarbeit ist es, einen Softwareprototyp zu entwickeln, der es ermöglicht, mit Hilfe der XLIFF-Technologie Software in der jeweiligen Landessprache anzuwenden, indem die Elemente der GUI in die verschiedenen Sprachen übersetzt werden. In dieser Arbeit steht deshalb die Programmierung und Darstellung des Prototyps sowie dessen Veranschaulichung im Benutzerhandbuch im Mittelpunkt. Dabei wird die XLIFF-Technologie näher betrachtet und deren Vorteile gegenüber der Ressourcen-Technologie aufgezeigt.

Den Abschluss der Bachelorarbeit bilden ein Leitfaden zur Einführung der XLIFF-Technologie in einem Softwareentwicklungsunternehmen und ein Ausblick hinsichtlich der Nutzung der XLIFF-Technologie.

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>VI</b>
<b>Tabellenverzeichnis</b>	<b>VII</b>
<b>Listings</b>	<b>VIII</b>
<b>Abkürzungsverzeichnis</b>	<b>IX</b>
<b>0 Einleitung</b>	<b>1</b>
0.1 Begriffe . . . . .	1
0.2 Aufgabenstellung . . . . .	2
0.3 Motivation . . . . .	3
0.4 Aufbau der Arbeit . . . . .	3
<b>1 IST-Analyse</b>	<b>4</b>
1.1 Was ist eine Ressource? . . . . .	4
1.2 Aufbau einer Ressourcendatei . . . . .	4
1.3 Ressourcentypen . . . . .	8
1.4 Zugriff und Verwendung . . . . .	8
<b>2 XLIFF</b>	<b>11</b>
2.1 Was ist XLIFF? . . . . .	11
2.2 Aufbau von XLIFF-Dateien . . . . .	12
2.2.1 XML . . . . .	13
2.2.2 XML-Schema . . . . .	13
2.2.3 XML-Schema Validation . . . . .	16
2.3 Anwendung in der Entwicklungsumgebung . . . . .	19
2.4 Zugriff und Verwendung . . . . .	19

<b>3</b>	<b>Vorteile und Nachteile</b>	<b>21</b>
3.1	Ressourcen-Technologie . . . . .	21
3.1.1	Vorteile . . . . .	21
3.1.2	Nachteile . . . . .	21
3.2	XLIFF-Technologie . . . . .	22
3.2.1	Vorteile . . . . .	22
3.2.2	Nachteile . . . . .	22
<b>4</b>	<b>Prototyp</b>	<b>23</b>
4.1	Aufgabenstellung . . . . .	23
4.2	Systemumgebung . . . . .	23
4.3	Programmierung . . . . .	24
4.3.1	Schema der Sprachänderung . . . . .	24
4.3.2	Umsetzung der Sprachänderung . . . . .	25
4.3.3	Setzen der lokalisierten GUI-Elementtexte . . . . .	26
4.4	Ressourcendatei . . . . .	28
4.5	XLIFF-Datei . . . . .	28
4.6	Test . . . . .	30
4.7	Benutzerhandbuch . . . . .	30
<b>5</b>	<b>Leitfaden</b>	<b>33</b>
<b>6</b>	<b>Zusammenfassung</b>	<b>35</b>
6.1	Ergebnis . . . . .	35
6.2	Ausblick . . . . .	36
	<b>Glossar</b>	<b>37</b>
	<b>Literaturverzeichnis</b>	<b>38</b>

# Abbildungsverzeichnis

1.1	Struktur einer Ressourcendatei . . . . .	5
1.2	Verwendung einer Ressource . . . . .	9
1.3	Zeitleiste des Betriebssystems Mac OS . . . . .	10
2.1	Schema einer XLIFF-Datei in der Anwendung . . . . .	12
4.1	Inhalt der Ressourcendatei des Prototyps . . . . .	28
4.2	Login-Fenster des Prototyps . . . . .	31

# Tabellenverzeichnis

1.1	Inhalt einer Ressourcendatei . . . . .	6
1.2	Struktur einer Ressourcendatei . . . . .	7
2.1	Attributwerte des file-Tags . . . . .	17
2.2	Werte des datatype-Attributs . . . . .	18
4.1	Gruppen der XLIFF-Datei . . . . .	29
4.2	Inhalt des Hauptmenüs . . . . .	32
4.3	Inhalt der Menüpunkte . . . . .	32

## Listings

2.1	Ausschnitt XML-Schema des xliif-Tags . . . . .	14
2.2	Ausschnitt XML-Schema des trans-unit-Tags . . . . .	14
2.3	Ausschnitt XML-Schema des source- und target-Tags . .	15
2.4	Beispiel eines XLIFF-Dokuments . . . . .	16
4.1	Quellcode zur Sprachermittlung . . . . .	25
4.2	Inhalt des XML-Dokuments Sprache.xml . . . . .	26
4.3	Quellcode zum Setzen der Texte in den GUI-Elementen	27
4.4	XLIFF-Dokument des Prototyps . . . . .	29

# Abkürzungsverzeichnis

DOM	Document Object Model
DTD	Document Type Definition
GUI	Graphical User Interface
OASIS	Organization for the Advancement of Structured Information Standards
XLIFF	XML Localization Interchange File Format
XML	Extensible Markup Language

## 0 Einleitung

In Zeiten der Globalisierung ist es wichtig, die Softwareentwicklung nicht nur national, sondern auch international zu forcieren. Um dieser Tatsache gerecht zu werden, ist eine Internationalisierung und Lokalisierung der Software notwendig.

Eine Möglichkeit dazu ist es, eine Software zu entwickeln, die diese Voraussetzungen erfüllt. Zur Verwirklichung dieses Vorhabens, der Lokalisierung von vertikalen Standardapplikationen mit Hilfe von XLIFF, hat der Autor einen Prototyp programmiert. Dabei wird unter Anwendung der XLIFF-Technologie die genutzte Software in die jeweilige Landessprache übersetzt.

Bei der sprachlichen Lokalisierung werden die einzelnen GUI-Elemente in die jeweilige Sprache übersetzt. Doch nicht nur die Oberfläche muss übersetzt werden, sondern auch die zum Programm dazugehörigen Dokumentationen. Die Dokumentationen in den verschiedenen Sprachen werden mit Hilfe einer separaten Gruppe innerhalb des jeweiligen XLIFF-Dokuments übersetzt.

### 0.1 Begriffe

#### **vertikale Standardapplikation<sup>1,2</sup>**

Vertikale Standardapplikationen, auch Standardanwendungen genannt, sind Anwendungen, die für einen bestimmten spezialisierten Markt entwickelt wurden. Diese Softwareanwendungen sind i.d.R. zweckgebunden und für einen bestimmten Bereich gedacht.

---

<sup>1</sup>Vgl. <http://www.entrepreneur-support.com/lang/de/starting-a-software-product-company.shtml> Abfrage am 16.08.2009

<sup>2</sup>Vgl. <http://home.arcor.de/hannesegger/VertikaleSoftware.html> Abfrage am 16.08.2009

Nachfolgend sind Beispiele für vertikale Standardapplikationen aufgeführt.

- Statistikprogramme für Architekten und Bauingenieure
- Honorarabrechnungsprogramme für Ärzte
- Finanzbuchhaltung für Kaufleute

## Lokalisierung

Der Begriff der Lokalisierung findet seine Verwendung im Bereich der Softwareentwicklung. "Als Lokalisierung bezeichnet man das Anpassen eines Programms an ein bestimmtes Land oder eine Region.

Zur Lokalisierung gehören das Übersetzen von Bildschirmtexten sowie das Anpassen von Datums-, Zeit- und Währungsdarstellungen. Der Aufwand für die Lokalisierung kann erheblich reduziert werden, indem nach dem Prinzip der Internationalisierung gearbeitet wird."<sup>3</sup> Bei diesem Prinzip werden alle sprachrelevanten Programmfragmente aus dem Programmcode extrahiert. Der Prozess der Lokalisierung umfasst vor allem die Auslagerung von landes- und regionsspezifischen Daten, wie Bildschirmausgaben und -texten.

## 0.2 Aufgabenstellung

Es soll ein Prototyp einer Softwareanwendung programmiert werden, in dem die GUI-Elemente mit Hilfe der XLIFF-Technologie lokalisiert werden. Außerdem sollen die XLIFF- und die Ressourcen-Technologie näher erläutert und deren Vor- und Nachteile aufgezeigt werden.

Dieser Prototyp wird unter der Plattform Microsoft Windows programmiert und auf den Plattformen Microsoft Windows und Apple Macintosh lauffähig sein.

Die Programmierung des Prototyps im Rahmen der Bachelorarbeit wird in dem Unternehmen Ingenious GmbH stattfinden.

---

<sup>3</sup>s. <http://www.bullhost.de/1/lokalisierung.html> Abfrage am 16.08.2009

## 0.3 Motivation

Die Lokalisierung von Softwareapplikationen ist ein sehr interessantes und faszinierendes Thema. Im Zuge eines anderen Softwareprojekts hat sich der Autor schon mit dem Thema der Lokalisierung und Internationalisierung von Software beschäftigt und dadurch einen Einblick in diese Thematik erhalten.

In der Bachelorarbeit soll dieses erworbene Wissen mit Hilfe der neuen vielseitigen und anspruchsvollen Übersetzungstechnologie XLIFF erweitert und vertieft dargestellt werden.

## 0.4 Aufbau der Arbeit

Im Kapitel IST-Analyse werden der derzeitige Stand der Ressourcen-Technologie, der Aufbau dieser Technologie sowie der Zugriff und die Verwendung von Ressourcendateien vorgestellt.

Im Kapitel XLIFF wird die XLIFF-Technologie im Bereich Lokalisierung und Internationalisierung in der Softwareentwicklung dargestellt. Es wird die XLIFF-Technologie, deren Aufbau, Zugriff und Verwendung erläutert.

Im Kapitel Vor- und Nachteile werden die Vor- und Nachteile der Ressourcen- und der XLIFF-Technologie gegenübergestellt.

Im Kapitel Prototyp werden die Herangehensweise bei der Programmierung des Prototyps und der im Projekt verwendeten XLIFF-Technologie erläutert sowie die Oberflächengestaltung anhand eines Demonstrators dargestellt. Der Demonstrator beinhaltet die Verwaltung von Kontakten, wobei Kontakte angelegt, geändert, gelöscht und gespeichert werden. Im Unterkapitel Test werden die Strategie und die Bedienung des Programms beim Prototyp-Test veranschaulicht.

Im Kapitel Leitfaden wird die Vorgehensweise zur Einführung der XLIFF-Technologie in einem Softwareentwicklungsunternehmen beschrieben.

Im Kapitel Zusammenfassung werden die Ergebnisse umrissen und ein Ausblick vorgenommen.

# 1 IST-Analyse

In diesem Kapitel werden der derzeitige Stand der Ressourcen-Technologie, der Aufbau dieser Technologie sowie der Zugriff und die Verwendung von Ressourcendateien vorgestellt.

## 1.1 Was ist eine Ressource?

”Eine Ressource ist Datenmaterial jeder Art, das in einem bestimmten Format in einer Windows Datei mit der Endung “.RSR” oder im Ressourcemark einer Macintosh Datei gespeichert ist. Ressourcen enthalten im Allgemeinen Strings, Bilder, Icons usw..”<sup>4</sup>

## 1.2 Aufbau einer Ressourcendatei

Eine Ressourcendatei ist stark strukturiert, enthält zusätzlich zu den Daten jeder Ressource eine Kopfzeile und eine Karte, die den Inhalt der Ressource vollständig beschreibt.

In der folgenden Abbildung wird die Struktur einer Ressourcendatei dargestellt und anschließend die einzelnen Segmente einer Ressourcendatei bzw. der Inhalt eines Ressourcemark aufgezählt.

---

<sup>4</sup>siehe [4D04] S. 1303

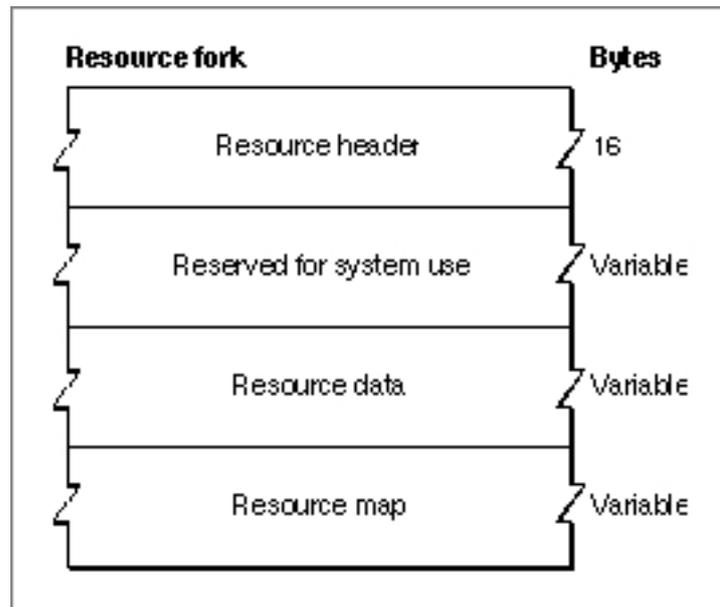


Abbildung 1.1: Struktur einer Ressourcendatei<sup>5</sup>

Diese Abbildung stellt eine typische Ressource einer Ressourcendatei oder des Resourceforks dar. Jede Ressource im System besitzt solch eine Struktur.

Diese Struktur beinhaltet

- einen Bereich, in dem die Größe der Ressource bestimmt (Resource header) wird,
- einen Bereich für das Betriebssystem,
- einen Bereich für die Daten (Resource data) und
- einen Bereich für die Ressourcenkennung (Resource map).

Beim Anlegen einer Ressource wird die Ressourcen-ID, der Ressourcentyp und der Ressourceninhalt übergeben.

In der nachfolgenden Tabelle werden die einzelnen Segmente einer Ressourcendatei oder eines Resourceforks näher erläutert.

<sup>5</sup>s. <http://developer.apple.com/documentation/mac/MoreToolbox/MoreToolbox-99.html> Abfrage am 14.09.2009

Segment	Beschreibung
Resource header	Im Resource header wird die Größe der Resource map und der Resource data definiert.
Reserved for system use	Dieser Bereich der Ressourcendatei oder des Ressourceforks ist für Instruktionen des Betriebssystems gedacht.
Resource data	Dieses Segment einer Ressourcendatei oder des Ressourceforks ist der wichtigste Teil. In diesem Segment werden die einzelnen Daten anhand des Ressourcentyps und der Ressourcen-ID bzw. des Ressourcennamens in der Ressourcendatei oder im Ressourcefork abgelegt. Das Thema Ressourcentypen wird in einen separaten Unterkapitel behandelt.
Resource map	In diesem Segment einer Ressource wird die Ressourcen-ID abgelegt. Diese ID muss eindeutig sein und darf nur einmal vorkommen.

**Tabelle 1.1:** Inhalt einer Ressourcendatei

Die nachfolgende Abbildung soll die Struktur einer Ressourcendatei unter dem Betriebssystem Windows veranschaulichen.

Die Struktur der Beispiel-Ressourcendatei ist in die Gruppen

- label,
- menu,
- title,
- button,
- help,
- table und
- language

unterteilt. Innerhalb der Gruppen sind die einzelnen Ressourcen untergebracht, die auf den einzelnen GUI-Elementen als Texte angezeigt werden.

Im Beispiel der Abbildung wurde nur der Ressourcentyp "STR#" genutzt.

Diese Struktur ist gleich der Struktur des XLIFF-Dokuments im programmierten Prototyp.

Gruppennr.	Ressourcenname	Ressourcenr.	Inhalt der Ressource
15001	label	1	Nutzer
		2	Passwort
		3	altes Passwort
		4	neues Passwort
		5	whlg. Passwort
		6	Name
		7	Straße / Hausnr.
		8	PLZ
		9	Ort
		10	Telefon
		11	Sprache
		⋮	⋮
15003	menu	1	Inhalt Ressource 1
		⋮	⋮
15004	title	1	Inhalt Ressource 1
		⋮	⋮
⋮	⋮	⋮	⋮

**Tabelle 1.2:** Struktur einer Ressourcendatei

Diese Ressourcendatei wird ebenfalls wie die erstellten XLIFF-Dokumente im Resources-Ordner des Datenbank-Projektes abgelegt.

Um die Ressourcendatei nutzen zu können, müssen seit der Version 11 der 4D Entwicklungsumgebung alle XLIFF-Dokumente aus dem Ordner Resources entfernt werden.

Die XLIFF-Technologie hat mit der Version 11 Vorrang.

## 1.3 Ressourcentypen<sup>6</sup>

Ressourcen sind nach Typen klassifiziert. Ein Ressourcentyp besteht aus einem String mit vier Zeichen. Es werden sowohl die Groß- und Kleinschreibung, als auch diakritische Zeichen berücksichtigt. Jede Art von Dateityp, seien es Textdateien oder Bilddateien, haben einen Ressourcentyp.

- Eine Ressource vom Typ "STR#" enthält eine Liste von Pascal Strings und kann diese in unbegrenzter Anzahl speichern. Diese Ressource heißt Stringlisten Ressource. Dieser Ressourcentyp ist der Standardtyp einer Ressource.
- Im Unterschied dazu, ist bei einer Ressource vom Typ "STR " das Leerzeichen als viertes Zeichen zu beachten. Diese Ressource heißt String Ressource und enthält einzelne Pascal Strings.
- Eine Ressource vom Typ "TEXT" enthält Text und kann diesen ohne begrenzte Textlänge speichern. Diese Ressource heißt Text Ressource und wird vor allem für längere Texte, die auf der Programmoberfläche erscheinen, genutzt, was hauptsächlich bei Fehlermeldungen auftritt.
- Eine Ressource vom Typ "PICT" enthält ein auf dem Macintosh basierendes QuickDraw Bild, das mit 4D unter den Betriebssystemen Windows und Apple Macintosh angezeigt werden kann. Diese Ressource heißt Bildressource und beinhaltet die Bild-ID des jeweils gewählten Bildes, womit auf die interne Bildbibliothek zurückgegriffen werden kann.

Oben genannte Liste enthält die wichtigsten Ressourcentypen, die auch im Rahmen der meisten Softwareentwicklungen genutzt werden. Neben den klassischen Ressourcentypen können auch eigene Ressourcentypen definiert werden.

## 1.4 Zugriff und Verwendung

Der Zugriff auf die Ressource in der Ressourcendatei oder im Ressourcefork erfolgt über die Ressourcen-ID bzw. über den Ressourcennamen, wenn keine Ressourcen-ID definiert worden ist. Der Zugriff auf eine Ressource wird auf einem Label, Button oder als Hilfstext dargestellt.

---

<sup>6</sup>siehe [4D04] S. 1306, 1307

Der Zugriff auf eine bestimmte Ressource kann beispielsweise wie folgt aussehen: Zugriff auf Ressource 15001,1. Es wird auf die 1. Ressource der Gruppe 15001 zugegriffen und diese auf einem Label, Button oder als Hilfstext dargestellt.

Jedes ausführbare Programm auf einen Apple Macintosh besteht aus jeweils einem Data- und einem Resourcefork.

Die Aufteilung in Forks hat den Vorteil, dass die Lokalisierung der Anwendungen leichter von statten gehen kann, da die wichtigen Daten, wie die Texte der GUI, extrahiert und vom eigentlichen Programmcode getrennt sind. Beide Forks sind in einer Datei zusammengefasst.

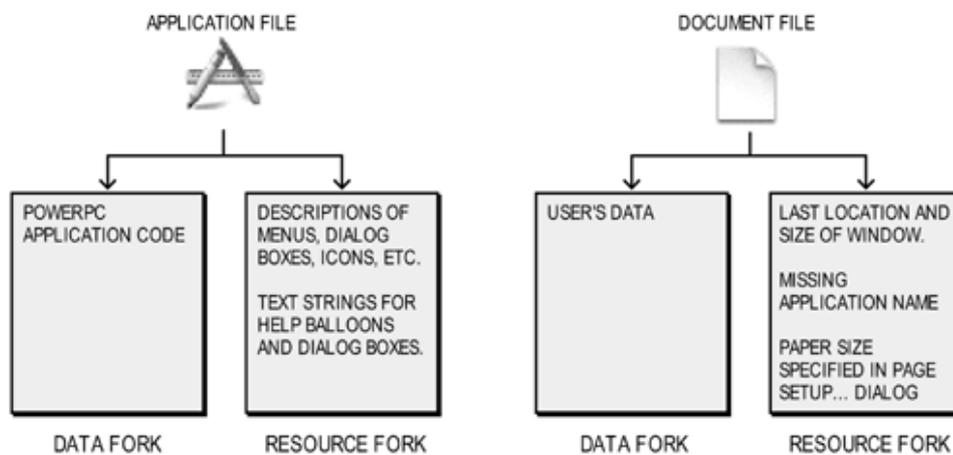


Abbildung 1.2: Verwendung einer Ressource<sup>7</sup>

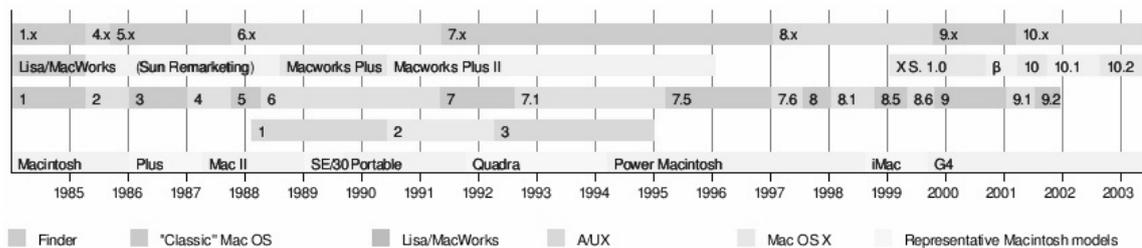
Die Ressourcen-Technologie von Apple findet nicht nur Anwendung im Bereich der Mac OS-Software, sondern auch jede Datei auf einen Mac OS-System hat einen Resourcefork. Die Resourcefork-Daten enthalten Informationen über die Datei, wie beispielsweise das Icon der Datei.

Ursprünglich wurden auf Macintosh Daten und Ressourcen in derselben Datei, bestehend aus Datafork und Resourcefork, gespeichert. Das Datafork einer Macintosh-Datei entspricht einer Datei unter Windows oder UNIX. Das Resourcefork einer Macintosh Datei enthält die auf Macintosh basierenden Ressourcen der Datei und hat keine direkte Entsprechung unter Windows und UNIX.

<sup>7</sup>s. <http://homepage.mac.com/mdouma46/dfont/dfont.html> Abfrage am 14.09.2009

Werden auf einem Macintosh-System Daten in eine Datei geschrieben und anschließend gespeichert, werden die gespeicherten Daten entweder in den Ressourcefork oder in den Datafork geschrieben. In der Regel werden Daten in den Datafork einer Datei abgelegt.

In der folgenden Abbildung wird die Zeitleiste der klassischen Mac OS-Versionen in Zeile 3 dargestellt.



**Abbildung 1.3:** Zeitleiste des Betriebssystems Mac OS<sup>8</sup>

Das System aus dem Datafork und dem Ressourcefork wurde bis zum Ende der klassischen Mac OS-Version 9 verfolgt. Mit der Nachfolgeversion von Mac OS-Version 9, der Mac OS-Version X, kann jeder Apple Entwickler die Ressourcefork-Daten in einer eigenständigen Datei ablegen. Diese Datei wird im gleichen Format wie die normalen Ressourcefork-Daten gespeichert.

In den aktuellen Macintosh-Anwendungen kommt die XLIFF-Technologie vorrangig zum Einsatz.

<sup>8</sup>s. <http://retromaccast.ning.com/photo/photo/show?id=1672786%3APhoto%3A45882> Abfrage am 25.09.2009, Abbildung in angepasster Form

## 2 XLIFF

In diesem Kapitel wird die XLIFF-Technologie im Bereich Lokalisierung und Internationalisierung in der Softwareentwicklung dargestellt. Es wird die XLIFF-Technologie, deren Aufbau, Zugriff und Verwendung erläutert.

### 2.1 Was ist XLIFF?

XLIFF ist eine Abkürzung und bedeutet XML Localization Interchange File Format und wird von dem Standardisierungsgremium<sup>9</sup> OASIS weiter entwickelt.

Die Version 1.0 wurde im Mai 2001 von einem Konsortium der führenden Unternehmen, wie IBM, Novell, Oracle, Sun Microsystems, der IT-Branche bereitgestellt. Damit wurde der Standard XLIFF geschaffen, der die zahlreichen Probleme im Management von mehrsprachigen Übersetzungen behebt.

Mit dem XLIFF-Standard werden elektronische Dokumente, Software und Webanwendungen lokalisiert. Dabei wird der Austausch von übersetzbaren Daten ermöglicht. Die XLIFF-Technologie ist der Industriestandard der Lokalisierung und löst damit die vielen unterschiedlichen Lokalisierungstechnologien ab.

XLIFF basiert auf dem XML-Standard und ist damit leicht auf viele Plattformen zu portieren und anwendbar. Die Dateierweiterung von XLIFF ist ".xlf".

Die aktuelle Version von XLIFF ist die Version 1.2.1 (s. [OAS07]).

In der nachfolgenden Abbildung wird die Vorgehensweise bei der Übersetzung mit der XLIFF-Technologie dargestellt. Das XLIFF-Dokument wird mit Hilfe eines XML-Parsers gelesen. Diese gelesenen XLIFF-Daten werden mit Hilfe der Ressourcen-ID oder des Ressourcennamens auf dem jeweiligen GUI-Element platziert.

---

<sup>9</sup><http://www.2was.de/nw/stand.pdf> Abfrage am 16.08.2009, siehe Glossar

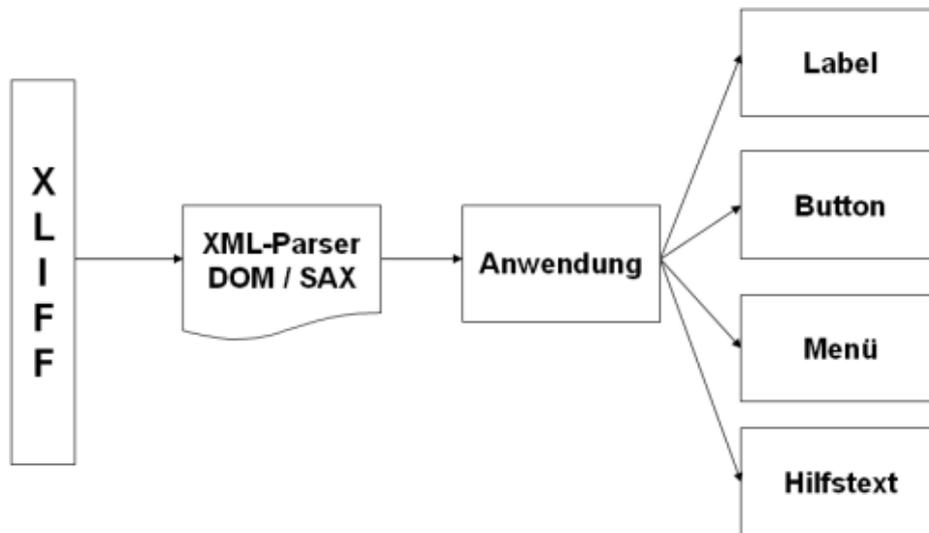


Abbildung 2.1: Schema einer XLIFF-Datei in der Anwendung<sup>10</sup>

Mit XLIFF können alle Elemente einer GUI lokalisiert werden. Dies umfasst die Beschriftung von statischen Textfeldern, wie

- Label,
- Button,
- Fenstertitel,
- Menüs mit Menütitel und Menüzeilen,
- Fehlermeldungsboxen oder
- sogenannte Tooltips (Hilfsteht).

Tooltips sind Schrifteinblendungen beim Überfahren eines Buttons oder bei Textfeldern.

## 2.2 Aufbau von XLIFF-Dateien

Um mit der XLIFF-Technologie arbeiten zu können, muss die Struktur der XLIFF-Dokumente verstanden werden.

---

<sup>10</sup>eigene Darstellung

Die XLIFF-Dokumente folgen einer strengen Struktur, um well-formed und valid zu sein. Damit ein XML-Dokument als valid bezeichnet werden kann, muss es wohlgeformt sein. Diese strenge Struktur der Wohlgeformtheit wird durch ein XML-Schema oder in einer etwas vereinfachten Form durch eine DTD definiert und beschrieben. Nach der Überprüfung auf Wohlgeformtheit ohne Fehler kann das Dokument als valid bezeichnet werden.

### 2.2.1 XML

Die XLIFF-Technologie basiert, wie bereits erwähnt, auf dem XML-Standard. Die Abkürzung XML steht für Extensible Markup Language und bedeutet erweiterbare Auszeichnungssprache. XML ist eine Metasprache. Sie beschreibt die Daten von bestimmten Dokumenten durch selbstdefinierte HTML-Tags. Im Gegensatz zu HTML trennt die Metasprache XML streng den Inhalt vom Layout. Bei diesem Vorgang ist zu beachten, dass die selbstdefinierten Tags auf der gleichen Hierarchieebene ein öffnendes und ein schließendes Tag besitzen. Die selbstdefinierten Tags werden in einem XML-Schema beschrieben.

### 2.2.2 XML-Schema

Ein XML-Schema erläutert ein XML-Dokument näher. Es werden die einzelnen Tags definiert, deren Attribute und Kindelemente. Ein XML-Schema erlaubt die Kontrolle darüber, welche Elemente ein Dokument enthalten darf und es ermöglicht differenzierte Einschränkungen in Bezug auf den Inhalt eines Elements.

In dem folgenden Quellcode wird das auf dem XML-Standard basierende XLIFF-Dokument definiert. Im Einzelnen wird das `xliff`-Tag beschrieben.

In Zeile 1 wird das Element-Tag `<xliff>` angelegt.

Nachdem dem Element ein Name gegeben wurde, wird dem Element in Zeile 2 ein komplexer Datentyp zugewiesen. Für das Element-Tag `<xliff>` wird der komplexe Datentyp verwandt, da das Element ein Attribut mit Namen `version` besitzt. Im XML-Schema wird zwischen einfachen und komplexen Datentypen unterschieden. Der Unterschied besteht darin, dass komplexe Datentypen Attribute und einfache Datentypen keine Attribute besitzen.

In Zeile 3 wird definiert, dass die folgenden Element-Tags unbegrenzt wiederholt werden können. Diesen Sachverhalt legt das Attribut `maxOccurs` mit dem Wert `unbounded` für unbegrenzt fest.

```

1 <xsd:element name="xliff">
2   <xsd:complexType>
3     <xsd:sequence maxOccurs="unbounded">
4       <xsd:any maxOccurs="unbounded" minOccurs="0" namespace="##other"
5         processContents="skip"/>
6       <xsd:element ref="xlf:file"/>
7     </xsd:sequence>
8     ... attributes ...
9 </xsd:complexType>
10 </xsd:element>

```

**Listing 2.1:** Ausschnitt XML-Schema des `xliff`-Tags<sup>11</sup>

Mit folgendem Quellcode wird eine Übersetzungseinheit in einem XML-Schema definiert. Jede Übersetzungseinheit beginnt mit dem Element-Tag `<trans-unit>` und wird auch damit beendet. Jedes `trans-unit`-Tag muss zwingend das Attribut `id` (Zeile 14) besitzen, um die Unterscheidung zwischen den einzelnen Übersetzungseinheiten zu ermöglichen.

Als weiteres wichtiges optionales Attribut kann zur Unterscheidung der Übersetzungseinheiten der Name der Ressource angegeben werden.

Innerhalb des öffnenden und schließenden `trans-unit`-Tags werden die beiden Element-Tags `<source>` und `<target>` als Referenzen definiert (Zeile 4 und 5).

Das Element-Tag `<source>` kann im Übersetzungseinheiten-Tag höchstens einmal oder überhaupt nicht auftreten. Hingegen das Tag `<target>` muss mindestens und maximal einmal vorhanden sein.

```

1 <xsd:element name="trans-unit">
2   <xsd:complexType>
3     <xsd:sequence>
4       <xsd:element ref="xlf:source"/>
5       <xsd:element minOccurs="0" ref="xlf:target"/>
6       <xsd:choice maxOccurs="unbounded" minOccurs="0">
7         ...
8       </xsd:choice>

```

<sup>11</sup><http://wiki.oasis-open.org/xliff/XLIFF1.2/Errata> Abfrage am 25.09.2009

```

9         <xsd:any maxOccurs="unbounded" minOccurs="0" namespace="##other"
10             processContents="strict"/>
11     </xsd:sequence>
12     <xsd:attribute name="id" type="xsd:string" use="required"/>
13     ... attributes ...
14 </xsd:complexType>
15 ...
16 </xsd:element>

```

**Listing 2.2:** Ausschnitt XML-Schema des trans-unit-Tags<sup>12</sup>

Mit den folgenden XML-Schema-Fragmenten werden die beiden Tags `<source>` und `<target>` definiert. Beide Elemente werden in der XML-Schema-Definition des Tags `<trans-unit>` per Referenz anhand des Elementnamens aufgerufen (Zeile 4 und 5). Die Referenz der beiden Element-Tags `<source>` und `<target>` greifen auf die Element-Definitionen zu.

Das Element-Tag `<source>` kann als Attribut den Wert `xml:lang` beinhalten. Dieses Attribut hat die Quellsprache zum Inhalt, kann aber auch weggelassen werden.

Die Quellsprache des Dokuments wird nach der ISO-Norm ISO639-1 als Länderkürzel verfasst und stellt standardmäßig jeden Ländernamen mit zwei Buchstaben dar. In dieser ISO-Norm steht für die deutsche Sprache die Abkürzung DE und für die englische Sprache die Abkürzung EN.

Das Element-Tag `<target>` kann optional Attribute beinhalten und muss im Übersetzungseinheiten-Tag einmal vorhanden sein. Neben dem Attribut `xml:lang` kann es die Attribute u.a. `font`, `style` besitzen. In dem optionalen Attribut `xml:lang` wird die Zielsprache in ISO-Norm ISO639-1 verfasst.

```

1 <xsd:element name="source">
2   <xsd:complexType mixed="true">
3     <xsd:group maxOccurs="unbounded" minOccurs="0" ref="xlf:
4       ElemGroup_TextContent"/>
5     <xsd:attribute ref="xml:lang" use="optional"/>
6     <xsd:anyAttribute namespace="##other" processContents="strict"/>
7   </xsd:complexType>
8   ...
9 </xsd:element>
10 <xsd:element name="target">
11   <xsd:complexType mixed="true">
12     <xsd:group maxOccurs="unbounded" minOccurs="0" ref="xlf:
13       ElemGroup_TextContent"/>

```

<sup>12</sup><http://wiki.oasis-open.org/xliff/XLIFF1.2/Errata> Abfrage am 25.09.2009

```

12     ... attributes ...
13   </xsd:complexType>
14   ...
15 </xsd:element>

```

**Listing 2.3:** Ausschnitt XML-Schema des source- und target-Tags<sup>13</sup>

### 2.2.3 XML-Schema Validation

Beim Vorgang der Validation per XML-Schema wird das XML-Dokument auf inhaltliche Korrektheit überprüft. Dabei werden die verwendeten Tags auf Gültigkeit im XML-Dokument geprüft. Prüfungen von XML-Dokumenten werden durch XML-Parser vollzogen.

Das gleiche Prinzip kann mit XLIFF-Dokumenten durchgeführt werden.

### Beispiel-Dokument

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <xliff version="1.2" xmlns="urn:oasis:names:tc:xliff:document:1.2" xmlns:sup="http
  ://www.ChaucerState.ac.pg/Frm/XLFSup-v1" xmlns:xsi="http://www.w3.org/2001/
  XMLSchema-instance" xsi:schemaLocation="http://www.ChaucerState.ac.pg/Frm/
  XLFSup-v1 s2-5-1.xsd">
3   <file datatype="x-STR#" source-language="de" target-language="en" product-
  version="2007">
4     <body>
5       <trans-unit id="1">
6         <source>Hallo Welt!</source>
7         <target>Hello World!</target>
8       </trans-unit>
9       <group id="15001">
10        <trans-unit id="1" resname="Hallo">
11          <source>Hallo Welt!</source>
12          <target>Hello World!</target>
13        </trans-unit>
14      </group>
15    </body>
16  </file>
17 </xliff>

```

**Listing 2.4:** Beispiel eines XLIFF-Dokuments<sup>14</sup>

<sup>13</sup><http://wiki.oasis-open.org/xliff/XLIFF1.2/Errata> Abfrage am 25.09.2009

<sup>14</sup>Quellcode aus dem Prototyp

Die nachfolgenden Erläuterungen beziehen sich auf das oben genannte Beispiel-Dokument.

## Generelle Struktur

Jedes XLIFF-Dokument beginnt mit einer klassischen XML-Deklaration. In dieser Deklaration steht `<?xml version="1.0"encoding="UTF-8"?>`. Nach der Deklaration wird der Dokumenttyp definiert.

Dabei wird gesagt, dass die Definition über die URL `http://www.ChaucerState.ac.pg/Frm/XLFSup-v1s2-5-1.xsd` bezogen werden soll.

Nachdem die Deklaration und der Dokumenttyp festgelegt sind, wird das eigentliche XLIFF-Dokument definiert. Jede XLIFF-Definition beginnt mit dem Tag `<xliff>` und wird erst am Ende des Dokuments mit dem End-Tag `</xliff>` beendet.

Das Start-Tag `<xliff>` enthält als Attribut die Version von XLIFF. Neben der Version 1.0 können auch die Versionen 1.1 und die aktuelle Version 1.2 als Attribut geschrieben werden. In der nächsten Zeile des XLIFF-Dokuments steht ein weiteres wichtiges Tag, das `file`-Tag. In diesem Tag wird festgelegt, wie der Datentyp sowie die Quell- und Übersetzungssprache benannt sind.

In der folgenden Tabelle werden die Attribute des `file`-Tags definiert.

---

Attribut	Inhalt
<code>datatype</code>	Dieses Attribut enthält den Datentyp der Begriffe. Im Beispiel-Dokument nimmt das Attribut den Wert <code>x-STR#</code> an. Das bedeutet, dass die Begriffe Stringlisten enthalten.
<code>source-language</code>	Dieses Attribut enthält die Quellsprache des XLIFF-Dokuments, welche die Ausgangssprache für die Übersetzung der Begriffe bildet.
<code>target-language</code>	Dieses Attribut enthält die Übersetzungssprache des XLIFF-Dokuments. Es ist die Zielsprache, in die die Begriffe übersetzt werden.

---

**Tabelle 2.1:** Attributwerte des `file`-Tags

Mit dem `datatype`-Attribut des `file`-Tags wird der Datentyp der Begriffe der Übersetzungseinheiten bestimmt.

In folgender Tabelle werden die anzunehmenden Werte des `datatype`-Attributs definiert.

Wert	Beschreibung
x-STR#	Dieser Wert enthält eine Stringliste.
STR	Dieser Wert enthält einen String.
PICT	Dieser Wert enthält Bilder.

**Tabelle 2.2:** Werte des `datatype`-Attributs

Im `source`-Tag einer Übersetzungseinheit steht der Begriff in der Quellsprache (Zeile 3, `source-language`). Im `target`-Tag einer Übersetzungseinheit steht der Begriff, der im `source`-Tag definiert wurde, in der Übersetzungssprache (Zeile 3, `target-language`).

## Übersetzungseinheit

Die Übersetzungseinheit beginnt mit dem Tag `<trans-unit>` und endet auch mit solch einem. Innerhalb dieses Tags gibt es ein Attribut `id`. Dieses Attribut ist eine eindeutige Zahlenwertzuweisung, die im Dokument nur einmal vorkommt.

Ist eine Übersetzungseinheit in einer Gruppe gekapselt, können sich die Übersetzungseinheiten-IDs gleichen und von Gruppe zu Gruppe wiederholen, allerdings pro Gruppe darf eine Übersetzungseinheiten-ID nur einmal vorkommen.

Die Übersetzungseinheiten haben zwei Kindelemente, den `source`-Tag (Zeile 6, 11) und den `target`-Tag (Zeile 7, 12).

Im `source`-Tag stehen ein bzw. mehrere Begriffe in der Quellsprache und im `target`-Tag stehen ein bzw. mehrere Begriffe in der Zielsprache.

Im `body`-Tag des Dokuments stehen die zu übersetzenden Begriffe. Diese Begriffe können einzeln nur im `body`-Bereich (Zeile 4-15) oder innerhalb einer gesonderten Gruppe stehen. Die einzelnen Begriffe werden in Übersetzungseinheiten zusammengefasst.

## Gruppen

Das `group`-Tag beinhaltet eine Zusammenfassung von Übersetzungseinheiten (Zeile 9-14). Das `group`-Tag besteht aus den Attributen `id` und `resname`. Im Attribut `id` steht ein eindeutiger Zahlenwert. Über diesen Zahlenwert und der `id` der Übersetzungseinheit wird der Begriff ausgelesen und auf der Anwendungsoberfläche angezeigt.

In einer Gruppe können unbegrenzt Übersetzungseinheiten vorhanden sein.

## 2.3 Anwendung in der Entwicklungsumgebung

In der 4th Dimension Entwicklungsumgebung der Firma 4D wird die XLIFF-Technologie seit der Version 11 unterstützt. In der Version 11 wird automatisch ein Ordner `Resources` im Datenbank-Projekt erzeugt, worin alle angelegten XLIFF-Dateien enthalten sind.

Die XLIFF-Dateien, die vorhanden sind, werden im Datenbank-Projekt automatisch geladen. Das bedeutet, dass nur die XLIFF-Daten der aktuellen Datenbank-Projekt-Sprache betroffen sind. Die Datenbank-Projekt-Sprache kann während der Laufzeit nicht geändert werden, was die Lokalisierung der Anwendung erschwert.

## 2.4 Zugriff und Verwendung

In der 4D Entwicklungsumgebung lassen sich die XLIFF Referenzen in verschiedenen Syntax-Schreibweisen verwenden.

- **15001,1**

Bei Verwendung dieser Syntax steht der Wert 15001 für das `group`-Element-Attribut `id`. Im Beispiel wird auf die Gruppe 15001 zugegriffen und der zweite Wert steht für die `id` der Übersetzungseinheit `trans-unit` der ausgewählten Gruppe. Im Beispiel wird auf die erste Übersetzungseinheit (`trans-unit`) zugegriffen und der Wert des Tags `<target>` angezeigt.

Diese Syntax ist identisch mit dem Zugriff auf die alten Ressourcen der Ressourcen-Technologie.

- **xliff:Hallo**

Bei Verwendung dieser Syntax wird auf das Attribut `resname` im Tag `<trans-unit>` zugegriffen. Das ist der Ressourcenname der Übersetzungseinheit und dieser Name muss einmalig im gesamten XLIFF-Dokument vorhanden sein.

Im Beispiel wird auf die Übersetzungseinheit mit dem Ressourcennamen `Hallo` zugegriffen und der Wert des Tags `<target>` angezeigt.

Dieser Syntax-Typ ist nicht kompatibel mit der alten Ressourcen-Technologie und kann daher nur mit geladenen XLIFF-Dateien funktionieren.

Die XLIFF-Technologie wird im Bereich der Softwareentwicklung in der Lokalisierung von GUI-Elementen angewendet. Sie wird vorrangig im Bereich der Desktop-Anwendungsentwicklung genutzt und findet auch Verwendung im Bereich der Webanwendungen.

## 3 Vorteile und Nachteile

In diesem Kapitel werden die Vorteile und Nachteile der Ressourcen- und der XLIFF-Technologie gegenübergestellt.

Die Verwendung von XLIFF-Dokumenten in der 4D Entwicklungsumgebung ist der Verwendung der Ressourcen-Technologie sehr ähnlich. Der Zugriff auf die jeweilige Ressource ist identisch. Beide haben den Zugriff über die Ressourcengruppen-ID und innerhalb der Gruppe über die Ressourcen-ID. Neben den String-Ressourcen können beide Technologien auch Referenzen auf Bilddateien beinhalten.

### 3.1 Ressourcen-Technologie

#### 3.1.1 Vorteile

1. Die Ressourcen-Technologie trennt die lokalisierbaren GUI-Elementtexte von dem eigentlichen Programmcode.
2. Die Ressourcen-Technologie erleichtert das Verwalten von Dokumentinformationen und das schnellere Anzeigen von Informationen.
3. Die Ressourcen-Technologie unterstützt neben den klassischen, siehe Kapitel 1.3, auch benutzerdefinierte Ressourcentypen.

#### 3.1.2 Nachteile

1. Die Verwendung der Ressourcen-Technologie erfolgt nur unter dem Betriebssystem Mac OS und ist dadurch auf andere Betriebssysteme schwer portierbar.
2. Die Ressourcendateien oder auch der Inhalt des Resourcefork kann nur von speziellen Ressourceneditoren gelesen und geändert werden. Solch ein Ressourceneditor ist beispielsweise ResEdit von Apple.

3. Die Ressourcen-Technologie ist in der Form der Aufteilung Datafork und Resourcefork veraltet und wurde nur bis zur Mac OS-Version 9 geführt.
4. Der Zugriff ist nur über die Ressourcengruppen-ID und über die Ressourcen-ID bzw. Ressourcenname, wenn keine Ressourcen-ID eingegeben, möglich.

## 3.2 XLIFF-Technologie

### 3.2.1 Vorteile

1. XLIFF ist Cross Platform-fähig. Das bedeutet, dass die XLIFF-Dokumente auf jedem gängigen Betriebssystem lesbar sind.
2. XLIFF-Dokumente können von jedem beliebigen Texteditor gelesen werden.
3. XLIFF ist ein Dialekt von XML und somit erfolgt eine komplette Trennung von Inhalt und Layout. Durch diese Eigenschaft ist die XLIFF-Technologie ein universell einsetzbares Dateiformat.
4. Die mehrsprachige Übersetzung wird durch Angabe von Quell- und Zielsprache und durch Eingabe von Quell- und Zielbegriff vereinfacht.
5. Neben dem Zugriff über die Gruppen- und Übersetzungseinheiten-ID ist auch der Zugriff über den Ressourcennamen möglich.

### 3.2.2 Nachteile

1. In großen Dokumenten ist es schwierig, auftretende Syntaxfehler zu finden.
2. Die XLIFF-Dokumente können sehr lang und unübersichtlich werden, wenn viele GUI-Elemente lokalisiert werden müssen.
3. Für jede einzelne Sprache muss ein separates XLIFF-Dokument existieren.

## 4 Prototyp

In diesem Kapitel werden die Herangehensweise bei der Programmierung des Prototyps und der im Projekt verwendeten XLIFF-Technologie erläutert sowie die Oberflächengestaltung anhand eines Demonstrators dargestellt. Der Demonstrator beinhaltet die Verwaltung von Kontakten, wobei Kontakte angelegt, geändert, gelöscht und gespeichert werden. Im Unterkapitel Test werden die Strategie und die Bedienung des Programms beim Prototyp-Test veranschaulicht.

### 4.1 Aufgabenstellung

Um die Lokalisierung von vertikalen Standardapplikationen mit Hilfe von XLIFF zu demonstrieren, hat der Autor einen Prototyp programmiert. Dieser Prototyp besitzt die Fähigkeit, Kontakte zu verwalten, indem sie angelegt, geändert, gelöscht und gespeichert werden können. Zusätzlich besteht die Möglichkeit, die Kontakte in einer Liste anzuzeigen. Die Software kann unter Microsoft Windows und auf Macintosh genutzt werden.

### 4.2 Systemumgebung

Die Programmierung des Prototyps erfolgte in der Ingenious GmbH, dem betreuenden Unternehmen. Die Ausstattung des Arbeitsplatz-PCs umfasste einen Fujitsu-Siemens-PC mit 1.7 GHz, 4 GB RAM, 500 GB Festplatte und ein DVD-Laufwerk. Als Entwicklungsumgebung wurde die Software 4th Dimension der Firma 4D GmbH genutzt.

## 4.3 Programmierung

Bei der Programmierung des Prototyps wurde der Schwerpunkt auf die sprachliche Lokalisierung gelegt. Die Lokalisierung mit Hilfe der XLIFF-Technologie erfolgte in den Sprachen Deutsch und Englisch. Dabei musste beachtet werden, dass alle GUI-Elemente (Labels, Buttons, Hilfstexte, Tabellenköpfe usw.) in der jeweilig eingestellten Sprache erscheinen.

Anhand der im Projekt verwendeten Projektmethoden wird die Nutzung der XLIFF-Technologie erläutert, wobei die wichtigste Funktion die Sprachänderung zwischen den beiden oben genannten Sprachen ist.

### 4.3.1 Schema der Sprachänderung

Im Ordner Preferences wird das XML-Dokument Sprache.xml ausgelesen. In diesem Dokument wird die zuletzt verwendete Spracheinstellung gespeichert. Dieses Dokument wird mit dem DOM-Parser eingelesen und die `source-language` und die `target-language` in Variablen gespeichert. Im Projekt ist die Quellsprache `source-language` immer Deutsch und nur die Zielsprache `target-language` muss geändert werden.

Folgendermaßen gestaltet sich der Ablauf der Lokalisierung der GUI im Prototyp.

1. Während des Startvorgangs wird die zuletzt eingestellte Sprache aus dem XML-Dokument Sprache.xml im Ordner Preferences ausgelesen. Sollte der Ordner Preferences und/oder das XML-Dokument nicht existieren, werden diese mit der Standardsprache Deutsch erstellt und das XML-Dokument Sprache.xml nochmals ausgelesen.
2. Mit Hilfe der ausgelesenen Sprache wird im Ordner Resources der Ordner [ausgelesene Sprache].lproj ausgewählt und das darin enthaltene XLIFF-Dokument gelesen.
3. Der nächste Schritt besteht darin, in dem gelesenen XLIFF-Dokument die Daten in der richtigen Gruppe und der richtigen Übersetzungseinheit für das jeweilige GUI-Element zu finden. Diese Daten werden mit Parametern übergeben.

- Die Lokalisierungssprache kann nur während des Logins geändert werden. Die Änderung erfolgt durch die Sprachauswahl innerhalb einer ComboBox. Bei diesem Vorgang werden das XML-Dokument Sprache.xml neu geschrieben, die Sprache neu ausgelesen und die GUI-Elemente erneut geladen.

### 4.3.2 Umsetzung der Sprachänderung

Beim Start des Programms wird das XML-Dokument Sprache.xml gelesen und die zuletzt eingestellte Sprache geladen. Danach wird das zur eingelesenen Sprache gehörige XLIFF-Dokument gelesen und die GUI auf die Sprache eingerichtet.

Im folgenden Listing wird das Auslesen der zuletzt eingestellten Sprache dargestellt.

In dieser Methode wird in Zeile 2 mit DOM-Befehlen zuerst das übergebene XML-Dokument geparkt und das Root-Element übermittelt.

In Zeile 3 wird anhand der XPath-Notation vom Root-Element das gesuchte Element übergeben. Wenn dieses Element existiert, wird an dieser Stelle angehalten.

Im vorletzten Schritt, Zeile 4, wird innerhalb des existierenden Elements nach dem Attribut `target-language`, der Zielsprache, in der das Programm lokalisiert wird, gesucht und in der Variablen `$lang` gespeichert.

Im letzten Schritt wird die Zielsprache als Parameter der Methode zurückgegeben.

```
1 $xpath:="/body/prototyp/language"
2 $docRoot:=DOM Parse XML source($folder)
3 $docElem:=DOM Find XML element($docRoot;$xpath)
4 DOM GET XML ATTRIBUTE BY NAME($docElem;"target-language";$lang)
5 $0:=$lang
6 DOM CLOSE XML($docRoot)
```

**Listing 4.1:** Quellcode zur Sprachermittlung<sup>15</sup>

Das nachfolgende XML-Dokument wird am Anfang des Programmstarts geladen und die Zielsprache in Zeile 4 im Tag `language` im Attribut `target-language` in einer Variablen gespeichert und dem Programm in Form einer Methoden-Parameter-Rückgabe übermittelt.

---

<sup>15</sup>Quellcode aus dem Prototyp

Kann das Programm das XML-Dokument nicht laden, wird das Dokument mit der Zielsprache Deutsch erstellt und erneut geladen. Beim Ändern der Lokalisierungssprache wird das XML-Dokument neu erstellt und erneut geladen.

```
1 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2 <body>
3   <prototyp>
4     <language source-language="de" target-language="en"/>
5   </prototyp>
6 </body>
```

**Listing 4.2:** Inhalt des XML-Dokuments Sprache.xml<sup>16</sup>

### 4.3.3 Setzen der lokalisierten GUI-Elementtexte

Eines der wichtigsten Themen bei der Programmierung von Programmen für den internationalen Markt ist die Umsetzung der Auslagerung von sprachrelevanten Programmfragmenten aus dem Programmcode und der Zugriff auf die sprachrelevanten Programmfragmente. Mit dieser Methode werden die Texte auf die Labels, Buttons, Menüs und Hilfstexte gesetzt.

In den ersten beiden Codezeilen des nachfolgenden Quellcodes werden die Parameter der Methode gesetzt. In Zeile 1 wird der Gruppenname und in Zeile 2 wird die Nummer der Übersetzungseinheit abgefragt.

In Zeile 3 des Listings wird das XLIFF-Dokument der jeweilig eingestellten Sprache als Parameter der Methode `XLIFF` ("FILE") übergeben. Wenn keine Datei übertragen wird, werden die Texte der Datenbank-Projekt-Sprache ausgegeben. Im vorliegenden Prototyp ist die interne Projekt-Sprache Deutsch.

Nach dem Parsen des XLIFF-Dokuments wird in Zeile 23 anhand des 1. Parameters der Methode in einer Gruppe, die von 1 bis 6 durchnummeriert ist, ermittelt und das Dokument bis zu dieser Gruppe durchlaufen. Wenn diese Gruppe existiert, wird das erste Kindelement der Gruppe ermittelt und aufgesucht.

Das Kindelement einer Gruppe ist die Übersetzungseinheit. Alle Übersetzungseinheiten einer Gruppe unterscheiden sich anhand einer ID im Attributbereich des Tags `<trans-unit>`.

---

<sup>16</sup>XML-Dokument des Prototyps

```
1 $docAttr:=$1
2 $docTransID:=$2
3 $xliffFile:=XLIFF ("FILE")
4
5 If (Length($xliffFile)>0)
6     If (Test path name($xliffFile)=Is a document )
7         $docRoot:=DOM Parse XML source($xliffFile)
8         If (OK=1)
9             Case of
10                : ($docAttr="label")
11                    $num:=1
12                : ($docAttr="menu")
13                    $num:=2
14                : ($docAttr="title")
15                    $num:=3
16                : ($docAttr="button")
17                    $num:=4
18                : ($docAttr="help")
19                    $num:=5
20                : ($docAttr="table"')
21                    $num:=6
22            End case
23            $docGroup:=DOM Find XML element($docRoot;"xliff/file/body/group["+
24                String($num)+"]")
25            If (OK=1)
26                $docTrans:=DOM Get first child XML element($docGroup)
27                If (OK=1)
28                    Repeat
29                        DOM GET XML ATTRIBUTE BY NAME($docTrans;"id";$id)
30                        If ($id=$docTransID)
31                            $docString:=DOM Get last child XML element(
32                                $docTrans)
33                            If (OK=1)
34                                DOM GET XML ELEMENT VALUE($docString;
35                                    $string)
36                                $0:=$string
37                            End if
38                        End if
39                    $docTrans:=DOM Get next sibling XML element(
40                        $docTrans)
41                Until (OK=0)
42            End if
43        End if
44    End if
45    DOM CLOSE XML($docRoot)
46 End if
```

**Listing 4.3:** Quellcode zum Setzen der Texte in den GUI-Elementen<sup>17</sup>

## 4.4 Ressourcendatei

Die nachfolgende Abbildung stellt eine Beispiel-Ressourcendatei des Prototyps dar. Diese Technologie wurde im Prototyp nicht verwandt und dient hier nur zur Veranschaulichung.

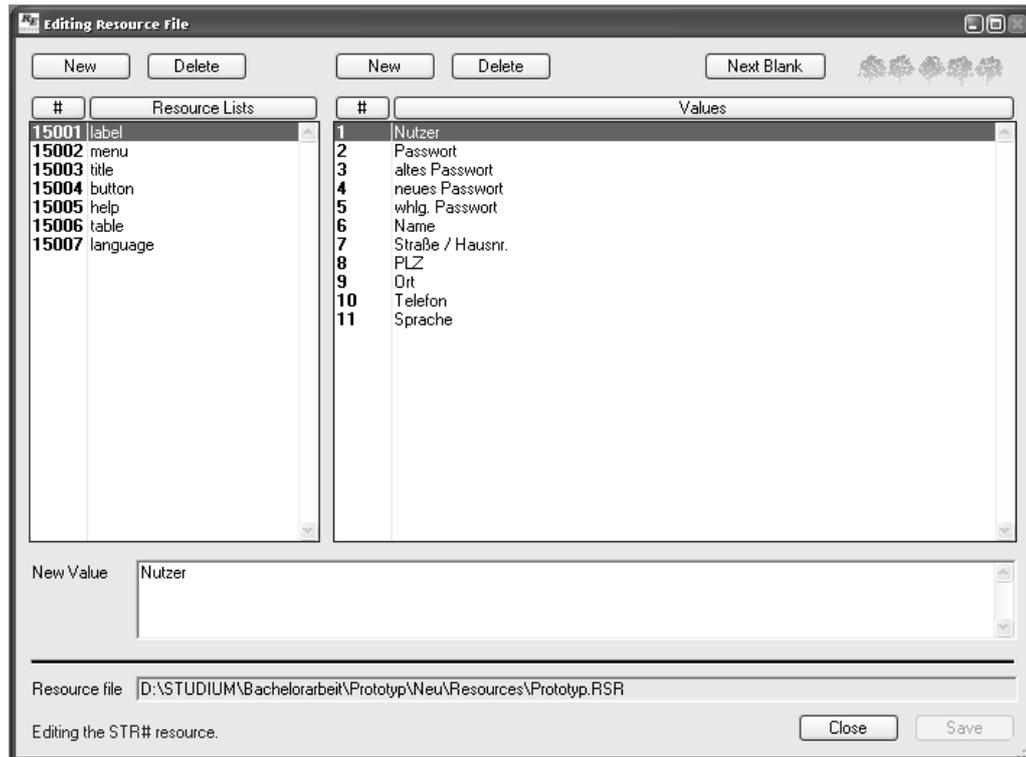


Abbildung 4.1: Inhalt der Ressourcendatei des Prototyps<sup>18</sup>

## 4.5 XLIFF-Datei

Die Struktur der zu erstellenden XLIFF-Datei wird in die Gruppen label, title, button, menu, help, table und language eingeteilt.

Diese Gruppen haben folgenden Inhalt.

<sup>17</sup>Quellcode aus dem Prototyp

<sup>18</sup>Beispiel-Darstellung des Ressourcendatei-Inhalts mit Hilfe des ResourceEditors 2.1

Gruppe	Inhalt
label	Fensterbeschriftungen
title	Fenstertitel-Beschriftungen
button	Button-Beschriftungen
menu	Menüs mit dem Menütitel und Menüzeilen
help	Fehlermeldungen
table	Tabellenkopf-Beschriftungen
language	Sprache in der jeweiligen Spracheinstellung

**Tabelle 4.1:** Gruppen der XLIFF-Datei

Der folgende Quellcode stellt die Grundstruktur der im programmierten Prototyp verwendeten XLIFF-Dokumente dar. In den XLIFF-Dokumenten werden zur Validierung DTDs genutzt, da in der 4D Entwicklungsumgebung die XLIFF-Version 1.0 verwandt wird.

Es handelt sich um das XLIFF-Dokument der Lokalisierungssprache Englisch im Ordner en.lproj. Wahlweise kann die Lokalisierungssprache Deutsch über den Ordner de.lproj verwendet werden.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE xliiff PUBLIC "-//XLIFF//DTD XLIFF//EN" "http://www.oasis-open.org/
   committees/xliiff/documents/xliiff.dtd">
3 <xliiff version="1.0">
4   <file datatype="x-STR#" source-language="de" target-language="en" product-
     version="2007">
5     <body>
6       <group id="15001" resname="label">
7         <trans-unit id="1">
8           <source>Nutzer</source>
9           <target>User</target>
10        </trans-unit>
11        ...
12      </group>
13      <group id="15002" resname="menu">
14        ...
15      </group>
16      <group id="15003" resname="title">
17        ...
18      <group id="15004" resname="button">
19        ...
20      <group id="15005" resname="help">

```

```
21     ...
22     <group id="15006" resname="table">
23     ...
24     <group id="15007" resname="language">
25     ...
26     </body>
27 </file>
28 </xliff>
```

**Listing 4.4:** XLIFF-Dokument des Prototyps<sup>19</sup>

## 4.6 Test

Am programmierten Prototyp wurde ein Komponententest durchgeführt, um Fehler aufzudecken und zu korrigieren.

Bei diesem Verfahren wurden die Projektmethoden getestet, die in direkter Berührung mit der XLIFF-Technologie stehen. Außerdem wurde überprüft, ob im Ordner Preferences automatisch bei Anwendungsstart und/oder die darin enthaltene XML-Dokument Sprache.xml erstellt wird, wenn der Ordner und/oder die Datei nicht vorhanden sind. Ein weiterer Test bestand darin, festzustellen, ob eine Fehlermeldung angezeigt wird, wenn das XLIFF-Dokument nicht im Sprachenordner vorhanden ist.

Zum Schluss der Testphase wurde ein Belastungstest durchgeführt. Damit wurde die Dauer der Laufzeit bei der Änderung der Lokalisierungssprache und der Anzeige der GUI-Elemente in der gewünschten Sprache getestet. Das Ergebnis des Belastungstests lag unter einer Sekunde. Alle Tests wurden erfolgreich absolviert.

## 4.7 Benutzerhandbuch

Der Prototyp zur Lokalisierung von Anwendungen auf der Grundlage der XLIFF-Technologie beinhaltet die exemplarische Verwaltung von Kontaktdaten. Das Verwalten besteht aus dem Anlegen, Ändern, Löschen und Speichern von Kontaktdaten. Die Kontaktdaten können in einer Liste dargestellt werden.

---

<sup>19</sup>Quellcode aus dem Prototyp

Zum Start des Prototyps erscheint ein Login-Fenster, in dem der Nutzernamen und das dazugehörige Passwort eingegeben werden müssen. Als Nächstes wird im Login-Fenster die Sprache, in der alle Labels, Buttons, Menüpunkte und Hilfstexte dargestellt werden sollen, ausgewählt. Zur Auswahl stehen die Sprachen Deutsch und Englisch. Sobald eine Sprache in der ComboBox festgelegt ist, ändert sich auch die Ansicht des Login-Fensters.

Die Lokalisierungssprache des gesamten Programms kann nur im Loginfenster des gestarteten Programms eingestellt werden.



Abbildung 4.2: Login-Fenster des Prototyps<sup>20</sup>

Nach dem Login wird das Hauptmenü des Prototyps angezeigt. Darin gibt es mehrere Auswahlmöglichkeiten zur weiteren Anwendung.

Auswahlmöglichkeiten:

- Kontaktliste anzeigen
- Über ...
- Programm beenden

In der nachfolgenden Tabelle werden die Auswahlmöglichkeiten erläutert.

---

<sup>20</sup>Screenshot vom Login-Fenster des Prototyps

<b>Button</b>	<b>Inhalt</b>
Kontaktliste anzeigen	In diesem Menüpunkt werden die Kontakte in einer Liste angezeigt. Es können Kontakte hinzugefügt, geändert und gelöscht werden.
Über ...	Es werden Informationen über den Verfasser und über die Gründe der Programmierung angezeigt.
Programm beenden	Bei Klick auf diesen Menüpunkt wird das Programm beendet.

**Tabelle 4.2:** Inhalt des Hauptmenüs

Die oben genannten Auswahlmöglichkeiten können durch Anklicken im Hauptmenü aufgerufen werden. Neben dem klassischen Hauptmenü können die oben genannten Funktionen auch über die Menüleiste im Menüpunkt Hauptmenü aufgerufen werden. Nachdem im Hauptmenü der Button Kontaktliste anzeigen geklickt wurde, wird in der Menüleiste ein weiteres Menü mit Namen Kontakte angezeigt.

Mit der folgenden Tabelle wird die Menüleiste mit den Menüpunkten und den Menüfunktionen erläutert.

<b>Menüpunkt</b>	<b>Funktion</b>	<b>Beschreibung</b>
Hauptmenü	Kontaktliste anzeigen	Es wird die gesamte Kontaktliste angezeigt.
Hauptmenü	Über ...	Es werden Informationen über den Prototyp angezeigt.
Hauptmenü	Neues Login anlegen	Es wird ein neues Login angelegt.
Hauptmenü	Passwort ändern	Passwort des aktuell eingeloggten Nutzers wird geändert.
Hauptmenü	Programm beenden	Das Programm wird beendet.
Kontakte	Kontakt anlegen	Neuer Kontakt wird angelegt.
Kontakte	Kontakt ändern	Der markierte Kontakt wird geändert.
Kontakte	Kontakt löschen	Der markierte Kontakt wird gelöscht.

**Tabelle 4.3:** Inhalt der Menüpunkte

## 5 Leitfaden

In diesem Kapitel wird die Vorgehensweise zur Einführung der XLIFF-Technologie in einem Softwareentwicklungsunternehmen beschrieben.

Dieser Leitfaden bezieht sich auf die Einführung von XLIFF in der 4D Entwicklungsumgebung und enthält folgende Schritte.

1. Im ersten Schritt wird eine Strategie zur Einführung der XLIFF-Technologie ausgearbeitet. Die Ausarbeitung beinhaltet die Finanzierung, das gegründete Projektteam und die Milestones.
2. Im zweiten Schritt wird eine Sicherung der vorhandenen Datenbankdaten vorgenommen.
3. Im dritten Schritt wird die neue Software beschafft. Dazu ist die Genehmigung der Projektfinanzierung notwendig.
4. Im vierten Schritt wird ein Testsystem mit der 4D Entwicklungsumgebung Version 11 aufgesetzt.

Es wird entweder die 4D Entwicklungsumgebung Version 11 oder die Version 2004 installiert, da entweder die komplette XLIFF, oder die komplette XML-Unterstützung benötigt wird. Die XML-Technologie wird ab der 4D Version 2004 unterstützt. Die Version 11 der Entwicklungsumgebung wird vorzugsweise verwendet, da sie die komplette XLIFF-Technologie enthält.

5. Im fünften Schritt wird ein Datenbank-Projekt im Testsystem, was den Ordner mit der Datenbankstruktur und den Datenbankdaten enthält, angelegt. Dieses Projekt kann an einer beliebigen Stelle auf der Festplatte abgelegt werden.
6. Im sechsten Schritt wird überprüft, ob im neu angelegten Datenbank-Projekt-Ordner die Unterordner Resources und Preferences existieren. Wenn sie nicht vorhanden sind, müssen sie angelegt werden.

Im Ordner Resources werden die XLIFF-Dokumente in den jeweiligen Sprachen abgelegt und im Ordner Preferences wird in einem XML-Dokument die zuletzt verwendete oder aktuell verwendete Sprache gespeichert.

7. Im siebten Schritt wird die 4D-Komponente 4D Pop installiert, um mit der XLIFF-Technologie und den XLIFF-Dokumenten besser arbeiten zu können. Diese Komponente kann kostenlos von der Website <http://www.4d.com> heruntergeladen werden.

Zur Installation der Komponente wird im Datenbank-Projekt ein Ordner Components angelegt und der heruntergeladene Inhalt wird in den Ordner kopiert.

8. Im achten Schritt wird mit Hilfe des XLIFF-Editors der Komponente 4D Pop ein XLIFF-Dokument der jeweiligen Systemsprache (i.d.R. in der jeweiligen Landessprache) angelegt. Bei diesem Vorgang wird im Ordner Resources ein Unterordner mit dem Namen de.lproj angelegt, in welchem das XLIFF-Dokument erstellt wird. Mit dem XLIFF-Editor werden die Gruppen und die Übersetzungseinheiten erzeugt.
9. Im neunten Schritt wird das einzuführende System getestet, indem die gesicherten Datenbankdaten in das Testsystem eingespielt werden. Zusätzlich werden weitere Tests, wie Komponententest, Belastungstest usw. durchgeführt.
10. Im zehnten Schritt wird das Testsystem durch die Anwender im Unternehmen geprüft.
11. Im letzten Schritt wird die derzeitige Entwicklungsumgebung durch das Testsystem, eine neue Entwicklungsumgebung, im Unternehmen ersetzt.

Während der Einführungsphase werden beide Entwicklungsumgebungen parallel geführt, um eventuelle Korrekturen der neuen Entwicklungsumgebung überbrücken zu können.

## 6 Zusammenfassung

In diesem Kapitel werden die Ergebnisse umrissen und ein Ausblick vorgenommen.

### 6.1 Ergebnis

In der Bachelorarbeit zum Thema "Lokalisierung von Standardapplikationen mit Hilfe von XLIFF" ist die Programmierung eines Prototyps mittels XLIFF-Technologie beschrieben, dargestellt und veranschaulicht worden.

Der Prototyp, dessen Aufgabe es ist, mit Hilfe der XLIFF-Technologie die GUI zu lokalisieren, wurde programmiert und die einzelnen Programmierphasen beschrieben.

Bei der Programmierung des Prototyps hat sich als Schwierigkeit herausgestellt, dass die aktuelle Lokalisierungssprache während der Laufzeit nicht umgewandelt werden kann. Dieser Zustand hat zur Folge, dass die Lokalisierungssprache während der Anwendung nur über spezielle Anpassungen, wie im Prototyp beschrieben, geändert werden kann.

Die Vorteile beim Einsatz der XLIFF-Technologie sind die unkomplizierte Portierbarkeit auf unterschiedliche Betriebssysteme sowie die einfache Anpassbarkeit der XLIFF-Dateien. Der Prototyp ist vor allem in Unternehmen, die die 4D Entwicklungsumgebung als Grundlage zur Programmierung nutzen, anwendbar.

Der Code kann auch in angepasster Form in anderen Programmiersprachen eingesetzt werden, da viele Programmiersprachen die XML- und XLIFF-Technologie unterstützen.

## 6.2 Ausblick

Die XLIFF-Technologie ist die Zukunft der Internationalisierung und Lokalisierung im Bereich der Softwareentwicklung.

Die flexible Einsetzbarkeit und hohe Anpassungsfähigkeit der XLIFF-Technologie sind gegenüber der Ressourcen-Technologie von enormen Vorteil.

Die XLIFF-Technologie wird sich als Standard, wie ihrerseits die XML-Technologie im Bereich Webdesign, durchsetzen.

Zur Zeit wird an der XLIFF-Version 2.0 gearbeitet. Der Termin für die Veröffentlichung dieser Version steht noch nicht fest. Laut der OASIS soll diese Version eine Steigerung für alle XLIFF-Nutzer werden.<sup>21</sup>

---

<sup>21</sup>Vgl. siehe May 2009: [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=xliff](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xliff)  
Abfrage am 25.09.2009

## Glossar

### **4D**

Das Softwareunternehmen 4th Dimension, auch 4D genannt, vertreibt eine Entwicklungsumgebung für Datenbankanwendungen. Die Firma wurde 1984 in Frankreich von Laurent Ribardière gegründet. Ursprünglich wurde die Entwicklungsumgebung für das Betriebssystem Mac OS von der Firma Apple entwickelt.

### **diakritische Zeichen**

Diakritische Zeichen (Diakritika, Einzahl Diakritikum) sind kleine Striche, Punkte, Häkchen etc., die meist über bzw. unter einem Basisbuchstaben stehen und seine Aussprache bzw. Betonung verändern. Sie dienen, wie schon ihr Name sagt, als unterscheidende Zeichen im Zusammenhang mit ansonsten identisch aussehenden Buchstaben.

### **Standardisierungsgremium**

Standardisierungsgremien sind Organisationen oder Vereine, die spezielle Standards und Normen für bestimmte Produktgruppen und technische Geräte bestimmen. Standardisierungsgremien sind u.a. ISO, OASIS, W3C, DIN.

### **valid**

Ein XML-Dokument ist valid, wenn das Dokument mit einer DTD oder einem XML-Schema konform ist.

### **well-formed**

Der Begriff well-formed kann mit wohlgeformt übersetzt werden und tritt vor allem in der Verbindung mit dem Begriff XML auf. Ein XML-Dokument ist beispielweise dann wohlgeformt, wenn es zu einem öffnenden Tag in der gleichen Hierarchie ein schließendes Tag gibt oder das Tag explizit als leeres Tag deklariert wird.

## Literaturverzeichnis

- [4D04] 4D: *4D Programmiersprache Handbuch Band 1/2/3*. 4D, 2004
- [App96a] APPLE, Inc.: *The Data Fork and the Resource Fork*. <http://developer.apple.com/legacy/mac/library/documentation/mac/MoreToolbox/MoreToolbox-11.html>. Version: 1996. – Abfrage am 14.09.2009
- [App96b] APPLE, Inc.: *Resource File Format*. <http://developer.apple.com/legacy/mac/library/documentation/mac/MoreToolbox/MoreToolbox-99.html>. Version: 1996. – Abfrage am 14.09.2009
- [App96c] APPLE, Inc.: *Resources*. <http://developer.apple.com/legacy/mac/library/documentation/mac/Toolbox/Toolbox-17.html#HEADING17-0>. Version: 1996. – Abfrage am 14.09.2009
- [App09] APPLE, Inc.: *Internationalization and Localization*. [http://developer.apple.com/mac/library/documentation/MacOSX/Conceptual/BPInternational/Articles/InternatAndLocaliz.html#//apple\\_ref/doc/uid/20000277](http://developer.apple.com/mac/library/documentation/MacOSX/Conceptual/BPInternational/Articles/InternatAndLocaliz.html#//apple_ref/doc/uid/20000277). Version: 2009. – Abfrage am 14.09.2009
- [Bal00] BALZERT, Helmut: *Lehrbuch der Software-Technik Software-Entwicklung*. Spektrum Akademischer Verlag, 2000
- [IBM02] IBM: *What is XLIFF?* <http://developers.sun.com/dev/gadc/technicalpublications/articles/xliff.html>. Version: 2002. – Abfrage am 14.09.2009
- [Mer03] MERCURY: *XLIFF*. <http://www.mercury-online.com/de/technologien/xliff.html>. Version: 2003. – Abfrage am 14.09.2009
- [OAS07] OASIS: *XLIFF 1.2 Specification*. <http://docs.oasis-open.org/xliff/xliff-core/xliff-core.html>. Version: 2007. – Abfrage am 14.09.2009
- [SL05] SPILLNER, Andreas ; LINZ, Tilo: *Basiswissen Softwaretest*. 3., überarbeitete und aktualisierte Auflage. dpunkt Verlag, 2005

# Selbständigkeitserklärung

Ich erkläre, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

*Mittweida, 15. Dezember 2009*

---

Gregor Gawol