
Ausarbeitung Austauschformate

Informationssysteme
5YHITM 2016/17

Maximilian Seidl

Note:
Betreuer: M.Martinides, C.Brein

Version 0.1
Begonnen am 19. Februar 2017
Beendet am 19. Februar 2017

Inhaltsverzeichnis

1	Aufgabenstellung	1
2	XML - eXtensible Markup Language	2
2.1	Herkunft	2
2.2	Metasprache	2
2.3	Dokumenttyp Definition (DTD)	2
2.4	Beispiel	2
3	UML	3
3.1	Herkunft	3
3.2	Modellierungssprache	3
3.3	Diagrammtypen	3
4	EDI	3
4.1	Herkunft	3
4.2	EDIFACT-Messages	4
5	Rich Text Format	4
5.1	Definition	4
5.2	Aufbau	4
6	JSON - JavaScript object notation	5
6.1	Definition	5
6.2	Aufbau	5
6.3	Elemente	5
6.4	Konzept	5
7	CSV - comma separated values	6
7.1	Definition	6
7.2	Verwendung	6
8	PostScript	6
8.1	Definition	6
8.2	Funktionsweise	6

9	HTML	6
9.1	Definition	6
9.2	Aufbau	7
10	Plain Text	7
10.1	Definition	7
10.2	Funktionsweise	7

1 Aufgabenstellung

Findet Informationen über zumindest 10 Industriestandards für Austauschformate heraus und fasst diese zusammen.

Gebt im Weitere praktische Beispiele an.

2 XML - eXtensible Markup Language

2.1 Herkunft

XML (eXtensible Markup Language) ist am 10. Februar 1998 vom W3C standardisiert worden. Es ist eine Metasprache zum Definieren von Dokumenttypen. Tolle Antwort, mit der sich gleich zwei neue Fragen ergeben: Was zur Hölle ist eine Metasprache und was sind Dokumenttypen?

2.2 Metasprache

Eine Metasprache ist eine Sprache zum Definieren von Sprachen. Hä? Was soll das denn heißen? Nun, das bedeutet, daß XML die Regeln zur Verfügung stellt mit denen sich dann Sprachen wie zum Beispiel HTML definieren lassen. Will meinen, daß sich mit XML jeder der Lust und Laune hat eine eigene Markup Sprache mit eigenen Tags (oder besser Elementen) erschaffen kann. Ob damit dann jemand etwas anfangen kann steht allerdings auf einem anderen Blatt.

2.3 Dokumenttyp Definition (DTD)

Um eine Sprache zu definieren, muß man ihre Grammatik und das Vokabular festlegen. Das tut man in der Dokumenttyp Definition (DTD), womit wir bei der zweiten Frage angekommen wären. Dokumente gehören zum selben Dokumenttyp, wenn sie der gleichen DTD folgen.

2.4 Beispiel

```
1      <?xml version="1.0"?>
      <BUCH>
3      <TITEL>Der Herr der Ringe</TITEL>
      <AUTOR>J.R.R. Tolkien</AUTOR>
5      </BUCH>
```

Listing 1: XML-Example

3 UML

3.1 Herkunft

UML ist im Zeitraum Ende der achtziger bis Anfang der neunziger Jahre entstanden mehrere Notationen zur Modellierung von OO-Systemen. Die Grundlage der Notationen waren objektorientierte Programmiersprachen, wie Smalltalk und C++.

3.2 Modellierungssprache

UML ist eine Modellierungssprache, also eine Sprache zur Beschreibung von Software-Systemen. Sie bietet eine einheitliche Notation, die für viele Anwendungsgebiete nutzbar ist. Sie enthält Diagramme und Prosa-Beschreibungsformen. Mit Hilfe der UML können statische, dynamische und Implementierungsaspekte von Softwaresystemen beschrieben werden

3.3 Diagrammtypen

UML enthält nachfolgende Diagrammtypen, welche in der Praxis eingesetzt werden.

- Strukturdiagramme (statische Aspekte)
 - Klassendiagramm - Class Diagram
 - Objektdiagramm - Object Diagram
 - Komponentendiagramm - Component Diagram
 - Paket Diagramm - Package Diagram
 - Kompositionsstrukturdiagramm - Composite Structure Diagram
 - Verteilungsdiagramm - Deployment Diagram
- Verhaltensdiagramme (dynamische Aspekte)
 - Aktivitätsdiagramm - Activity Diagram
 - Anwendungsfalldiagramm - Use Case Diagram
 - Zustandsdiagramm - State Machine Diagram
 - Interaktionsdiagramme - Interaction Diagram

4 EDI

4.1 Herkunft

EDI (Electronic Data Interchange) bezeichnet den Austausch strukturierter Daten zwischen Geschäftspartnern. Es handelt sich um einen Oberbegriff (nicht um eine bestimmte Norm, wie oft vermutet wird) und ist nicht zu verwechseln mit dem Begriff EDIFACT. Um EDI zwischen Partnern mit unterschiedlichen IT-Systemen zu ermöglichen, gab es immer wieder Normierungs-Bestrebungen.

Gängige EDI-Normen sind z.B. EDIFACT, ODETTE, VDA oder auch XML-EDI (letzteres in vielen Varianten wie z.B. ebXML, OpenTrans, Zugferd, RosettaNet oder das neue UNIDOC®TM aus dem Hause EDICENTER).

4.2 EDIFACT-Messages

Tabelle 1: EDIFACT Messages

ORDERS	purchase order	Bestellung (Auftrag)
ORDCHG	purchase order change request	Bestelländerung
ORDRSP	purchase order response	Auftragsbestätigung
DESADV	despatch advise	Lieferavis (Lieferschein)
INVOIC	invoice and credit note	Rechnung und Gutschrift
DELFOR	delivery schedule	Lieferabruf / Lieferplan
IFTMIN	instruction for transport	Transport- / Speditionsauftrag / Frachtbrief
IFTSTA	status of transport	Statusnachricht zu einer Lieferung
CUSDEC	customs declaration	Zollerklärung
PRICAT	price catalogue	Preisliste / Artikelstammdaten
SLSRPT	sales report	Abverkaufsbericht
INVRPT	inventory report	Lagerbestandsbericht
BAPLIE	stowage plan for container-vessel	Stauplan für Containerschiffe
APERAK	application error and acknowledgement	Statusrückmeldung auf Anwendungsebene
PAYMUL	multiple payment order	Zahlungsaufträge
FINSTA	financial statement of an account	Bankkontoauszug

5 Rich Text Format

5.1 Definition

Mit dem Rich Text Format (RTF) können ASCII-Texte mit Steuerzeichen codiert werden. Dieses von Microsoft entwickelte Dateiformat eignet sich für den Austausch von Texten zwischen Textverarbeitungsprogrammen unter Beibehaltung von Formatierungszeichen.

5.2 Aufbau

Der im RTF-Dateiformat gespeicherte Text kann mit einem anderen Textverarbeitungsprogramm wieder geöffnet werden und entspricht dem Originaltext. Beim RTF-Dateiformat werden Markups, die in geschweifte Klammern gestellt werden, in den Text eingefügt um damit den Inhalt mit bestimmte Befehlen zu codieren, Sonderzeichen einzufügen, die Attribute eines Absatzes zu definieren und die Funktionalität vom RTF-Format zu erhöhen. und damit die er Inhalt und durch bestimmte Befehle codiert. Zu den Attributen und Funktionalitäten gehören die Hyperlinks, Wertetabellen, horizontale Linien, Bookmarks, Änderungskennzeichen, Papierformate, der Satzaufbau, Bilder, die Kopf- und Fußzeilen und die Ausgabe. Die Extension von RTF-Dateien ist *.rtf.

6 JSON - JavaScript object notation

6.1 Definition

Die JavaScript Object Notation (JSON) ist ein kompaktes Format zum Austausch von Daten, welches von Douglas Crockford definiert wurde. Die Basis bildet dabei eine Teilmenge der JavaScript Programmiersprache - konkret die JavaScript-Notation für Objektliterale - gemäß des ECMA-262 Standards (Dritte Edition, 1999). Trotzdem handelt es sich bei JSON um eine Vorgehensweise, die Daten sprachunabhängig auf einfache Weise strukturiert und die sich inzwischen für viele weitere Programmiersprachen wie Java, Programmiersprache C, C++ u.v.a. durchgesetzt hat.

6.2 Aufbau

Damit kann JSON verwendet werden, um Daten zwischen Programmen auszutauschen, die in verschiedenen Programmiersprachen implementiert wurden. Man spricht in diesem Zusammenhang auch von der Serialisierung von Daten. Bei JSON handelt es sich um ein von Mensch und Maschine gleichermaßen zu lesendes Textformat. Der große Vorteil von JSON liegt sowohl in der einfachen Handhabung wie auch Implementierung.

6.3 Elemente

JSON unterscheidet Elemente wie Objekte, Arrays, Strings, Zahlen, Boolesche Werte (true und false) sowie den Spezialwert 0. Vor oder hinter jedem Element können Leerzeichen, Tabulatoren, Carriage Return und Newline-Zeichen, in Summe sogenannte Whitespaces, stehen. Die ausführliche Verwendung von Whitespaces ist hinsichtlich der Aspekte Übertragungs- oder Speicherkosten abzuprüfen.

6.4 Konzept

Der eigentliche Grundgedanke von JSON ist, dass diese Zeichenfolgen (Objektliterale) sehr einfach wieder in JavaScript-Arrays oder -Objekte zurück transformiert werden können. Da JSON selbst gültigen Java-Script-Code repräsentiert, kann die Zeichenfolge direkt ausgeführt und somit in ein JavaScript-Objekt überführt werden. Wird die Zeichenfolge jedoch mit der in JavaScript vordefinierten eval()-Funktion transformiert, so kann beliebiger JavaScript-Code ausgeführt werden, und somit auch unsicherer Code in die eigentliche Applikation eingestreut werden. Diese Gefahr lässt sich jedoch vermeiden, wenn man die JSON.parse-Methode zur Transformation (json2.js) statt eval() verwendet. Alternativ lässt sich auch für die Verwendung von JSON in JavaScript ein Framework wie mootools einsetzen, welches saubere Funktionen zur Analyse von Datenstrukturen (Parsen) und zur Wandlung von Objekten nach JSON bietet.

7 CSV - comma separated values

7.1 Definition

Comma Separated Values (CSV) ist ein Textformat, das für den Import von Texten in Spreadsheets und SQL-Datenbanken genutzt wird, ebenso wie für den Export aus SQL-Datenbanken. Beim CSV-Format werden die einzelnen Texte einer Zeile durch Kommas oder Semikolon getrennt.

7.2 Verwendung

Das CSV-Dateiformat ist eine Variante des ASCII-Zeichensatzes und des TXT-Formats für tabellarische Daten und den Austausch von reinen Daten.

8 PostScript

8.1 Definition

PostScript ist eine Seitenbeschreibungssprache, die in den frühen 1980er Jahren von Adobe Systems entwickelt wurde. Sie wird üblicherweise als Vektorgrafikformat für Dokumente und Drucker verwendet, stellt jedoch auch eine Turing-vollständige, stackorientierte Programmiersprache dar. PostScript ist eine Weiterentwicklung von Interpress.

8.2 Funktionsweise

Grafiken und Druckseiten werden als Dateien im PostScript-Format angelegt, um sie auf den unterschiedlichsten Ausgabegeräten in beliebiger Größe und Auflösung verlustfrei ausgeben zu können. Dazu werden grafische Elemente und Schriften als skalierbare Vektorgrafik beschrieben. Rastergrafiken können ebenfalls eingebettet werden; sie werden je nach Auflösung des Ausgabegeräts neu skaliert.

PostScript ist eine Turing-vollständige Programmiersprache. Sie ist stapelorientiert und funktioniert nach dem Prinzip der umgekehrten polnischen Notation. Vorbild war die Programmiersprache Forth. Postscriptfähige Ausgabegeräte (insbesondere Drucker und Druckmaschinen) sind mit einem Raster Image Processor (RIP), also einem Interpreter auf Hardware- oder Software-Basis, ausgestattet, der das PostScript-Programm Stück für Stück auswertet und in eine Rastergrafik umsetzt (siehe auch Postscript-Druckraster). Durch diese sequenzielle Befehlsausführung hat man in PostScript-Dateien keinen direkten Zugriff auf einzelne Seiten. Eine freie Software-Implementierung eines solchen Interpreters bietet die Software Ghostscript.

9 HTML

9.1 Definition

Hypertext Markup Language (HTML) ist die Beschreibungssprache für Dokumente im World Wide Web. Sie ist abgeleitet von der Standardized Generalized Markup Language (SGML) und

enthält die Sprachelemente für den Entwurf von Hypertext-Dokumenten und die Einbindung von Multimedia-Objekten.

9.2 Aufbau

Um die Komponenten, die Hierarchie und die Verknüpfungspunkte der Dokumente zu beschreiben, benutzt HTML HTML-Codes oder HTML-Tags. HTML hat einige Instrumente um einen direkten Verweis auf Animationen oder Sound-Dateien mittels Markups durchzuführen. Mit den HTML-Tags können einfache typografische Textauszeichnungen wie die Schriftausrichtung (mittig, links bündig, rechts bündig), die Schriftgröße und die Schriftausführung (fett, normal, unterstrichen, kursiv) vorgenommen werden. Desweiteren gibt es HTML-Befehle für den Absatzaufbau und den Zeilenumbruch, für Aufzählungen, Verweise, für die Formatierung und Positionierung von Grafiken und Fotos sowie für farbige Flächen.

10 Plain Text

10.1 Definition

Ein Klartext, Clear Text oder Plaintext, ist ein unverschlüsselter, direkt lesbarer Text, der voll ausformuliert sein kann. Durch Verschlüsselung des Klartextes mit dem Schlüssel des Verschlüsselungsalgorithmus wird der Klartext zum Geheimtext, durch Entschlüsselung des Geheimtextes wird dieser zum Klartext.

10.2 Funktionsweise

Klartexte setzen sich aus den Buchstaben und Ziffern des Klartextalphabets zusammen, mit denen der Klartext verschlüsselt wird. Es handelt sich in der Regel um das normale Alphabet mit einer endlichen Anzahl an Buchstaben und Ziffern. Bei Strichcodes wird häufig unter dem Strichcode eine Klartextzeile eingefügt. Diese enthält die Nutzdaten des Strichcodes, in manchen Fällen auch die Prüfziffer, allerdings keine Steuerzeichen. Die Klartextzeile sollte eigentlich für die optische Zeichenerkennung in OCR-Schrift sein, ist meistens aber in einer normalen Schrift ausgeführt. Die Klartextzeile kann für die manuelle Zeicheneingabe benutzt werden, wenn die gescannte Eingabe des Strichcodes fehlerhaft ist, oder vom Computer nicht angenommen wird.

Abbildungsverzeichnis