Software & Schnittstellen

Bachelor Informationsmanagement
Modul Digitale Bibliothek (SS 2014)

Dr. Jakob Voß

2014-05-19

Software

- Computerprogramm zur automatischen Ausführung ausgewählter Aufgabe
- auch bekannt als: App, Skript, Tool, System...
- auch verbunden mit Hardware
- auch aus verschiedenen Programmen (Modulen) bestehend
- auch basierend auf anderen Programmen

Beispiel: Smartphone-Betriebssystem & Apps

Software in (digitalen) Bibliotheken

Beispiele?!

Software in (digitalen) Bibliotheken

- Browser
- Repository-Software
- Integriertes Bibliothekssystem
  - Katalog
  - Erwerbungsmodul
  - Ausleihmodul
  - ...
- Discovery-Interface
- ...
- Personenverzeichnis o.A. Normdatenbank
- Crawler
- Datenkonvertierungsskripte
- ...

Software vs. Daten

- Software besteht aus Regel
  (Wenn X, dann Y...)
- Software wird ausgeführt
  (Erst A, dann B...)
- Software besteht aus Daten
- Software kann Daten lesen und erzeugen
  (über Schnittstellen!)

Software oder Daten?

- Beispiel: Personenverzeichnis
- Beispiel: Schema

Schnittstellen

Alles, worüber Daten in ein Programm herein und/oder herauskommen

Allgemeine Arten von Schnittstellen

- User Interface (UI)
  Benutzeroberfläche
- Application Programming Interface (API)
  Programmierschnittstellen

APIs

- Ermöglichen die Nutzung von Diensten und Methoden durch andere Programme
- Programme “sprechen” miteinander über APIs
  - falls sie die gleiche API nutzen (Spezifikation)
  - und diese gleich interpretieren (Implementation)

Eine API ist wie eine kleine, sehr spezielle Sprache

Kompatibilität

Kompatibilität durch genaue Einhaltung der Spezifikation

Spezifikation durch Standards (“Grammatik & Vokabular”)
Implementation durch Programme (“Aktiv- & Passive Sprachfähigkeit”)

Grundaufbau einer API

- Daten rein (Anfragre), Daten raus (Antwort)
- meist aufgeteilt in Server & Client

Einfaches Beispiel: HTTP-Anfrage/Antwort

...

Weniger einfaches Beispiel: OAI-PMH

OAI Protocol for Metadata Harvesting (OAI-PMH)

Data-Provider z.B. Server für Wissenschaftliche Schriften der Hochschule Hannover (SerWisS)

Service-Provider z.B. Bielefeld Academic Search Engine (BASE)

mehr siehe Veranstaltung am 24.3.2014

Mögliche Anfragen (verbs) per OAI-PMH

Identify	Was ist das hier für ein Repository?
GetRecord	Gib mit Metadatensatz!
ListRecords	Gib mir alle Metadatensätze!
ListIdentifiers	Welche Identifier gibt es?
ListMetadataFormats	Welche Metadatenformate gibt es?
ListSets	Welche Sets gibt es?

Einfacheres Beispiel: OpenSearch Suggest

siehe Live-Demo (Browser & Suggest-Dienste der VZG)...

Wo könnte sowas in digitalen Bibliotheken relevant sein?

Welche Dienste/Einrichtungen/... sollten OpenSearch Suggest anbieten?

Noch ein Beispiel: unAPI

Zweck Bereitstellung einzeln identifizierter digitaler Objekte

Anfragesyntax HTTP-Request mit zwei Parametern

- `id`: Identifikator des Objektes
- `format`: Gewünschtes Format

Antwortsyntax Formatliste in XML oder digitales Objekt in einem gewünschten Format

Beispiel: unAPI

- `BASEURL`: Liste von allgemeinen Formaten
- `BASEURL?id=IDENTIFIER`: Liste von Formaten für ein ausgewähltes Objekt
- `BASEURL?id=IDENTIFIER&format=FORMAT`: ausgewähltes Objekt in ausgewähltem Format

APIs und Datenformate

- APIs legen gewisse Datenformate fest (meist zumindest die Datenstrukturierungssprache wie z.B. JSON)
- APIs lassen oft gewissen Datenformate oder Felder offen
  - HTML, JSON, Bilder, Videos... über HTTP
  - DC, MARC... über unAPI oder OAI-PMH
  - ...
- APIs können aber auch sehr eingeschränkt und speziell sein
  - OpenSearch Suggest (JSON über HTTP)

Wie entstehen APIs?

Beispiel: Entwicklung der Document Availability Information API (DAIA) für Beluga und GBV-Bibliotheken

Henne-Ei-Problem: ohne APIs keine neuen Anwendungen

Was tun ohne APIs?

- Kapitulation
- Frickelei

Beispiel: Screen-Scraping

Übergangslösung: Wrapper

Verschiedene APIs für digitale Bibliotheken

Suchen	Z39.50, SRU/SRW, OpenSearch
Aggregation	OAI-PMH, ATOM, RSS, Sitemaps, ResourceSync
Statusabfragen	unAPI, SeeAlso, DAIA
Änderung	SRU Update
Benutzerkonto	PAIA

und viele (oder wenige) andere mehr

Arten von APIs

- Lesezgriff vs. Lese/Schreibzugriff
- offen vs. intern
- standardisiert vs. ad-hoc
- zustandslos vs. zustandsbehaftet
- Webservices vs. APIs über andere Protokolle

Zustandslose APIs

- Anfragen unabhängig voneinander in beliebiger Reihenfolge
- Eine Anfrage pro Aktion (sonst: Transaktionen)

Gegenbeispiel: Formular ausfüllen in mehreren Schritten

Grundsätzliche best practices für APIs

- **standardisiert**
- **abgrenzbarer Zweck**[1]
- möglichst **zustandslos**[2] oder mit Transaktionen
- **offen**
- **Webservices**

[1] don't trust the eierlegende Wollmilchsau!

[2] oft als so genannte REST-APIs

Offene Schnittstellen

- Grundsätzlich freier Zugriff
- Beschränkung i.d.R. als Teil der API (z.B. Accounts)
- Dokumentiert und möglichst in Form von Programmbibliotheken implementiert

Mashups & Serviceorientierte Architektur (SOA)

- Anwendungen, die aus mehreren Komponenten per API zusammengebaut sind (wie Lego)
- Motivation zur Erstellung und Pflege von APIs

Weiteres Beispiel: PAIA

- Zugriff auf Benutzerkonten

Gut zu wissen

- APIs sind notwendige “Sprachen” für den Datenaustausch von Programmen
- Je mehr und einfachere APIs, desto flexibler
- Große Systeme und Anbieter tendieren dazu sich abzuschotten