





Ostfalia - Hochschule für angewandte Wissenschaften  
Fachbereich Wirtschaft  
Studiengang Wirtschaftsinformatik

# **Design von Hypermedia-APIs**

Bachelorarbeit

Zur Erlangung des Grades eines Bachelor of Science  
der Fakultät Wirtschaft  
der Ostfalia - Hochschule für angewandte Wissenschaften

eingereicht bei	Prof. Dr. XYZ Zweitprüfer XYZ
von	Marius Brederlow Pappelweg 1 38536 Meinersen OT Ahnsen Mat.-Nr. 70047338

Meinersen, den 14. Mai 2013

## **Zusammenfassung**

ca. 12 Zeilen

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation und Zielsetzung . . . . .	1
<b>2</b>	<b>Hypermedia</b>	<b>2</b>
2.1	Definition . . . . .	2
2.2	H-Faktoren nach Amundsen . . . . .	2
2.3	Richardsons Maturity Model . . . . .	7
2.4	Hypermedia in der Praxis . . . . .	7
2.5	Design-Prozess . . . . .	7
2.6	Anwendungsbeispiel . . . . .	7
<b>3</b>	<b>Bewertung</b>	<b>8</b>
3.1	Bewertungsgrundlage . . . . .	8
<b>4</b>	<b>Hypermedia-Formate</b>	<b>9</b>
4.1	Basisformate . . . . .	9
4.2	JSON . . . . .	9
4.2.1	Beispielantworten aus der Anwendung . . . . .	9
4.2.2	Bewertung JSON . . . . .	9
<b>5</b>	<b>Fazit und Ausblick</b>	<b>10</b>
	<b>Literaturverzeichnis</b>	<b>11</b>
	<b>Ehrenwörtliche Erklärung</b>	<b>12</b>

# 1 Einleitung

## 1.1 Motivation und Zielsetzung

Das Internet ist wohl das größte und umfassendste Informationssystem unserer Zeit. Der Erfolg des Internet beruht sicherlich auf der Tatsache, dass es möglich ist verschiedenste Informationen rund um den Globus miteinander zu vernetzen und abrufbar zu machen. Heutzutage kann sich niemand mehr ein Internet ohne RSS-Feed, Podcasts, Youtube und diverse andere Multimediale Inhalte vorstellen. Zudem ist das Internet per Definition ein hochgradig verteiltes System. Dieser Erfolg ist in erheblichem Umfang den verschiedensten Hypermedia-Formaten zuzuschreiben. Viele große und hochgradig skalierte Webanwendungen laufen auf der Basis von Hypermedia oder bieten zum Teil Hypermedia APIs als öffentliche Schnittstellen zu ihrem Dienst an. Einige bekannte Vertreter sind z.B. Amazon, Facebook und Twitter. Mit der Einführung von Representational State Transfer (REST) als ein Architekturansatz für verteilte Informationssysteme im Jahre 2000 durch Roy Fielding, gewann das Thema Hypermedia in den letzten Jahren auch für kleinere Unternehmensanwendungen zunehmend an Bedeutung. Ein wesentlicher Punkt bei der Umsetzung von REST als Architekturstil ist Hypermedia. In der Dissertation von Fielding beschreibt er einen Ansatz, den er selbst "Hypermedia as the engine of application state"(HATEOAS) nennt. Grundslegend geht es hierbei darum, Hypermedia zu verwenden um Informationen auszutauschen und den Status der Anwendung zu ändern. Dabei sollte das Hypermediaformat selbsterklärend und möglichst unabhängig vom Transportprotokoll sein. Von selbsterklärend im Kontext von Hypermedia spricht man, wenn das Hypermediaformat semantische und kontextbezogene Informationen mit sich bringt.

Für den Begriff Hypermedia selbst gibt es keine Norm oder Standardisierung, die beschreibt, was Hypermedia überhaupt umfassen muss. Daher tummeln sich am Markt verschiedenste Hypermediaformate mit unterschiedlichsten Charakteristiken und Möglichkeiten. Diese Arbeit hat zum Ziel eine Beurteilungsgrundlage für Hypermediaformate zu schaffen, verschiedene Formate miteinander zu vergleichen und auf ihre Leistungsfähigkeit hin zu untersuchen.

- Erläuterung des IST-Zustandes und
- der daraus resultierenden Probleme <sup>1</sup>

---

<sup>1</sup>(Vgl. Hesse, 1992, S. 11)

## 2 Hypermedia

### 2.1 Definition

Für den Begriff Hypermedia existiert keine offizielle Definition oder Norm. Diese Tatsache gestaltet den Vergleich von verschiedenen Formaten recht schwierig, da zuerst definiert werden muss, welche Merkmale und Funktionen Hypermedia und die Formate eigentlich ausmachen. Die ge-läufigste Form von Hypermedia ist sicherlich HTML in Verbindung mit dem Internet und Millionen von Websites. Hypermedia auf Websites zeichnet sich auf den ersten Blick durch 2 wesentliche Merkmale aus. Websites bestehen zum einen aus multimedialen Inhalten und sind zum anderen untereinander mit Verweisen verknüpft. Außerdem gibt es die Möglichkeit Daten an den Server über Formulare zu senden. Diese Merkmale lassen sich dem bekanntesten Hypermediaformat, (X)HTML ohne weiteres zuschreiben.

Hypermedia kann seine Anwendung nicht nur in Form des Internets finden. Die Kombination aus Internet, Webserver, HTTP als Transportprotokoll und dem Browser als Client ist die am weitesten verbreitete Anwendung von Hypermedia. Die Betrachtung von verschiedensten Hypermediaformaten soll innerhalb dieser Arbeit jedoch völlig losgelöst von Programmiersprachen, Technologien und serverseitigen Implementationen erfolgen. Spricht man von Hypermedia im Rahmen einer Unternehmensanwendung müssen diverse Merkmale genau betrachtet werden. Zwecks einer Definition und dem Bilden einer Bewertungsgrundlage für diese Arbeit werden im folgenden 2 Modelle vorgestellt. Diese Modelle sollen eine Grundlage schaffen um die Formate vergleichbar zu machen und zu definieren, was wesentliche Merkmale von Hypermediaformaten sind.

### 2.2 H-Faktoren nach Amundsen

Der Hypermediagedanke entstand aus der Fragestellung, wie man ein gutes verteiltes Informationssystem entwerfen kann. Grundsätzlich gibt es verschiedene Ansätze um ein verteiltes System zu entwerfen und die Komponenten miteinander kommunizieren zu lassen. Die zu gründe liegende Fragestellung ist hier, wie kann der Server private Objekte exportieren um sie für den Client sichtbar und benutzbar zu machen. Hierfür gibt es unterschiedliche Plattform- und sprachspezifische Lösungen. Die grundsätzliche Funktionsweise ist hierbei immer ähnlich. Die Objekte werden serialisiert über das Netzwerk geschickt und so an den Client zur Verarbeitung übergeben. Der Client selbst muss also über ein Verständnis der Daten verfügen. Diese Herangehensweise hat den Nachteil, dass eine Änderung der serialisierbaren Objekte auch eine Änderung des Clients zur Folge hat. Somit entsteht immer ein gewisser Grad an Kopplung zwischen Client und Server.

Amundsen beschreibt mit Hypermedia einen anderen Lösungsansatz. Eine Synchronisation von Client und Server, bei der private Datentypen miteinander geteilt werden ist nicht der richtige An-

satz. Stattdessen sollte eine Technik zur Datenbeschreibung unabhängig von Plattform, Sprache und privaten Datentypen verwendet werden. Diese Herangehensweise nennt Amundsen die Hypermedia Lösung. Die Probleme des Typemarshalling werden im Fall von Hypermedia durch das Anreichern der Nachrichten mit zwei Arten von Metadaten gelöst. Ein Teil sind Metadaten über die Daten selbst. Der andere Teil sind Metadaten über den Applikationsstatus und die Möglichkeiten für den Client, den Applikationsstatus zu ändern. Losgelöst von privaten Datentypen ermöglicht dieses nachrichtenorientierte Design, eine spezifischere Anpassung an die Anforderungen der Problemdomäne. Durch Hypermedia Nachrichten werden somit keine privaten Datentypen geteilt sondern vielmehr ein generelles Verständnis der Daten.<sup>2</sup>

Nach Amundsen besteht eine Hypermedia Nachricht somit aus insgesamt 3 Teilen:

- Daten
- Metadaten über die Daten
- Metadaten über den Applikationsstatus

Um Hypermedia selbst zu identifizieren, definiert Amundsen eine Sammlung abstrakter Eigenschaften und Funktionen, wie ein Hypermediaformat diese 3 charakteristischen Datenteile umsetzt. Diese Sammlung wird Hypermediafaktoren (H-Faktoren) genannt und ist grundsätzlich unabhängig von der verwendeten Plattform und dem Transportprotokoll. In der Regel wird bei Hypermedia-Applikationen HTTP als Transportprotokoll eingesetzt, weil es eine Reihe von Vorteilen wie z.B. Cache-Control bereits implementiert hat. HTTP ist aber keine zwingende Voraussetzung für den Einsatz von Hypermedia. Diese H-Faktoren unterteilen sich dann noch in die 2 Bereiche Links und Control Data. Diese insgesamt neun H-Faktoren bilden das Gerüst für ein beliebiges Hypermediaformat. So kann man anhand dieser Faktoren bestimmen, ob ein bereits existierender Medientyp den Applikationsspezifischen Anforderungen genügt oder aber was bei der Erstellung eines eigenen Medientyps beachtet werden muss.<sup>3</sup>

Im Folgenden werden die einzelnen Faktoren genauer betrachtet. Die Link-Faktoren dienen im Wesentlichen der Bewegung des Clients im Applikationsfluss.

Embedding Links(H-Faktor LE) dient der Darstellung einer Ressource im aktuell genutzten Ausgabefenster. Ein Embedding Link nutzt die Lesen-Operation des verwendeten Protokolls und stellt die Antwort innerhalb des aktuellen Fensters dar. Ein gebräuchliches Beispiel für Embedding Links sind auf einer Website eingebundene Bilder.<sup>4</sup>

Outbound Links (H-Faktor LO) sind das, was man im Allgemeinen unter Navigationslinks versteht. Beim Folgen eines Outbound Links wird die Lesen-Operation des verwendeten Protokolls ausgeführt und die bestehende Ansicht durch die Antwort ersetzt.<sup>5</sup> Die Antwort von GOOGLE auf eine Suchanfrage beinhaltet Outbound Links. Folgt man einem Verweis aus der Antwort wird die darauf

---

<sup>2</sup>(Vgl. Amundsen, 2012, S. 10)

<sup>3</sup>(Vgl. Amundsen, 2012, S. 13f)

<sup>4</sup>(Vgl. Amundsen, 2012, S. 15)

<sup>5</sup>(Vgl. Amundsen, 2012, S. 15)

folgende Antwort die aktuelle Ansicht ersetzen.

Templated Links (H-Faktor LT) haben wie Embedding und Outbound Links nur einen lesenden Charakter. Templated Links reichern eine lesende Anfrage mit zusätzlichen Informationen an. Beispielsweise bei einem Suchdienst kann die Anfrage so erweitert werden, dass der Suchbegriff als ein Teil des Links mit in der Anfrage verarbeitet wird.<sup>6</sup>

Idempotent Links (H-Faktor LI) haben einen schreibenden Charakter und stellen eine Möglichkeit dar, Ressourcen auf dem Server anzulegen oder zu löschen. Idempotenz hängt hierbei natürlich von dem verwendeten Transportprotokoll und der serverseitigen Implementation ab. Bei der Verwendung von HTTP als Transportprotokoll sind PUT und DELETE idempotente Operationen zur Kommunikation mit dem Server.<sup>7</sup>

Non-Idempotent Links (H-Faktor LN) bietet wie H-Faktor LI eine Möglichkeit Daten zu übertragen. Der Unterschied hier ist die Idempotenz. So liegt es in der Verantwortung des Medientyps zu definieren, wie mit nicht idempotenten Operationen umgegangen wird. Im Fall von HTTP ist die Methode POST als nicht idempotente Operation implementiert. Sie wird beispielsweise genutzt um per Formular eine neue Ressource auf dem Server anzulegen.<sup>8</sup>

Die Control Faktoren sind im wesentlichen dadurch definiert, dass sie zusätzliche Informationen beim Ausführen eines Link zu Verfügung stellen. Hier geht es vor allem um die Beschreibung der beinhalteten Nutzdaten. Die Berücksichtigung dieser Faktoren trägt dazu bei, dass alle Seiten etwas über die Beschaffenheit der Daten wissen oder aber die Daten in einer gewünschten Kodierung anzufordern.

Read Controls (H-Faktor CR) beschreibt ob und wie die Möglichkeit besteht, eine lesende Operation zu reglementieren. Ein Beispiel hierfür sind die Accept-Header des HTTP Protokolls. Dem Client wird so die Möglichkeit gegeben ein bestimmtes Format oder die Antwort in einer bevorzugten Sprache anzufordern.<sup>9</sup>

Update Controls (H-Faktor CU) bietet die gleichen Möglichkeiten, die der H-Faktor CR bietet für schreibende Operationen. Die Reglementierung von schreibenden Operationen kann bei HTTP z.B. durch das Content-Typ Feld im Header durchgeführt werden. So hat der Server Kenntnis darüber, in welcher Form ihm die Nutzdaten übergeben werden.<sup>10</sup>

Method Controls (H-Faktor CM) stellen Indikatoren dar, die beschreiben um was für eine Art von Operation es sich handelt bzw. welche Operation für einen konkreten Fall gültig sind. Ein HTML FORM-Element kann mit unterschiedlichen Methoden abgesendet werden. Das Attribut METHOD beschreibt an dieser Stelle mit welcher Operation des Transportprotokolls das Formular übertragen wird.<sup>11</sup>

---

<sup>6</sup>(Vgl. Amundsen, 2012, S. 16)

<sup>7</sup>(Vgl. Amundsen, 2012, S. 16)

<sup>8</sup>(Vgl. Amundsen, 2012, S. 17)

<sup>9</sup>(Vgl. Amundsen, 2012, S. 18)

<sup>10</sup>(Vgl. Amundsen, 2012, S. 18)

<sup>11</sup>(Vgl. Amundsen, 2012, S. 18f)



Zusätzlich zu den 9 H-Faktoren definiert Amundsen 4 Design-Elemente. Diese Design-Elemente stellen vor allem vier charakteristische Eigenschaften von Hypermedia Formaten dar. Aus diesem Grund werden sie im Folgenden beschrieben und im weiteren Verlauf zum Vergleich unterschiedlicher Formate verwendet.

Die vier Hypermedia Design-Elemente sind: <sup>12</sup>

- Base Format
- State Transfer
- Domain Style
- Application Flow

Jeder Hypermediatyp basiert auf einem bestimmten zugrundeliegenden Format. Bekannte Formate sind XML, JSON, CSV und viele weitere. Ein Hypermediatyp kann auf einem beliebigen Basisformat aufbauen. Erst die Anreicherung des Basisformats mit zusätzlichen Metadaten und der Unterstützung von Hypermedia-Faktoren in einem bestimmten Umfang definieren einen Medientyp. Der Medientyp bestimmt außerdem ob eine Zustandsänderung der Applikation unterstützt wird oder nicht. Man kann hier unterscheiden zwischen der Art und Weise wie ein Zustandsänderung durchgeführt werden kann. Hierbei gibt es 3 Ausprägungen der Unterstützung. Keine Unterstützung bei nur lesenden Formaten. Definierte Unterstützung nach einer externen Dokumentation und Ad-Hoc Unterstützung mittels einer in die Nachricht eingebetteten Hypermedia Steuerung. Der Domain Style beschreibt wie stark ein Medientyp an eine Problemdomäne angepasst ist. Diese Stile sind kategorisiert als spezifisch, allgemein und agnostisch. Als letztes Design Element gilt es noch den Application Flow zu erwähnen. Hierbei geht es darum ob und in welchem Maße der Client in der Lage ist den Applikationsfluss zu steuern. Auch hier unterscheidet man 3 Ausprägungen. Der Applikationsfluss kann entweder nicht, wesentlich oder hauptsächlich durch den Medientyp gesteuert werden. <sup>13</sup>

Zusätzlich zu den Hypermedia-Faktoren und den Design Elementen definiert Amundsen noch 4 Ebenen auf denen sich ein Hypermedia Typ befinden kann. Diese vier Ebenen sind aus der Sicht des Clients beschrieben und unterscheiden sich nach ihren Möglichkeiten, in wie fern sie im Falle einer Veränderung angepasst werden müssen. Sie sind also gekennzeichnet durch einen bestimmten statischen und einen variablen Anteil. Statisch bezieht sich in diesem Kontext darauf, dass die statischen Anteile bei einer Veränderung oder Erweiterung der Applikation angepasst werden müssen. Die variablen Anteile hingegen können verändert werden ohne Probleme bei der Kommunikation zwischen Client und Server zu verursachen. Entweder dem statischen oder varia-

---

<sup>12</sup>(Vgl. Amundsen, 2012, S. 20)

<sup>13</sup>(Vgl. Amundsen, 2012, S. 20)

blen Anteil zugeordnet werden die vier Eigenschaften Content, Address, Read Write Semantics und Appflow. Diese Eigenschaften sind wie folgt definiert:<sup>14</sup>

- Content beschreibt die Fähigkeit Elemente einer Nachricht hinzuzufügen oder zu entfernen.
- Address ist definiert als die Möglichkeit, einen Unified Resource Identifier zu ändern, an den der Client seine Anfragen sendet.
- Read Write Semantics ist eine Eigenschaft, bezüglich des Schreib-Leseverhaltens. Ein Hypermediatyp, der diese Eigenschaft besitzt ist in der Lage anzuzeigen, welche Elemente einer Nachricht schreibbar sind und welche Protokolfunktion zu benutzen sind.
- Appflow bezeichnet die Möglichkeit den Applikationsfluss kontextbezogen zu steuern. Der Client benötigt nur die Antwort des Servers und kann die Steuerelemente der Nachricht benutzen um sich im Applikationsfluss zu bewegen.

Als Ebene 0 definiert Amundsen Serialisierte Objekte. Diese Ebene trifft auf Techniken wie SOAP und RPC-XML zu. Die Kommunikation zwischen Client und Server läuft über ein spezielles, Anwendungsbezogenes Nachrichtenformat. Die Definition der Kommunikationsregeln, also welches Format mit welchem Inhalt wo benutzt wird muss durch eine externe Dokumentation erfolgen. Diese Regeln werden dann im Client codiert und sie verfügen über keinerlei variablen Anteil. Die Konsequenz ist, dass der Client im Falle einer Änderung in allen 4 Eigenschaften angepasst werden muss.

Ebene 1 ist beschrieben als standardisierte Datenformate. Beispielhafte Medientypen sind Comma Separated Value (CSV) oder Extendible Markup Language (XML). Die Kommunikation zwischen Client und Server läuft über ein standardisiertes Datenformat. Dies hat den Vorteil, dass beide Seiten in der Lage sind den Nachrichten zu validieren. So können Elemente oder Attribute einer Nachricht hinzugefügt oder entfernt werden und die Kommunikation läuft weiterhin fehlerfrei. Der variable Anteil beschränkt sich jedoch auf die Eigenschaft Content. Die weiteren drei Eigenschaften sind nach wie vor statisch und eine Änderung würde zu Fehlern in der Kommunikation führen.<sup>15</sup>

Ebene 2 erweitert die standardisierten Datenformate mit Links. Atom und AtomPub sind Medientypen dieser Ebene. Der wesentliche Unterschied ist, dass ab dieser Ebene URIs in den Nachrichten enthalten sind. URIs werden somit nicht mehr im Client codiert sondern der Client bekommt mit jeder Nachricht URIs geliefert, denen er folgen kann. Dieser Fakt erweitert auch den variablen Anteil aus Ebene 1 um die Eigenschaft Address. Eine Anwendung, die einen Medientyp dieser Ebene benutzt kann ein variables URI-Schema benutzen. Sollte der Server seine URIs ändern, kann er dies tun ohne die Kommunikation zu bestehenden Client zu gefährden. Die Eigenschaften Read Write Semantics und Appflow sind weiter statisch.<sup>16</sup>

<sup>14</sup>Vgl. Amundsen, M. (2010), <http://amundsen.com/hypermedia/scraps/#hypermedia-levels>

<sup>15</sup>Vgl. Amundsen, M. (2010), <http://amundsen.com/hypermedia/scraps/#hypermedia-levels>

<sup>16</sup>Vgl. Amundsen, M. (2010), <http://amundsen.com/hypermedia/scraps/#hypermedia-levels>

Ebene 3 ist genauso definiert wie Ebene 2, jedoch erweitert um die Möglichkeit der Applikationssteuerung. Die Kopplung zwischen Client und Server, die einen Medientyp dieser Ebene benutzen, besteht aus einer Vereinbarung über den verwendeten Datentyp und den Elementen einer Nachricht, die zur Applikationssteuerung benutzt werden. Der Client muss in der Lage sein die Steuerelemente einer Nachricht zu erkennen und dem Benutzer so die derzeit aktuellen Steueroptionen anzuzeigen. Hypermediaformate dieser Ebene sind unter anderem die Hypertext Markup Language (HTML) und VoiceXML.<sup>17</sup>

Betrachtet man die H-Level nach Amundsen etwas abstrakter, sind die Ebenen nach dem Grad der Kopplung zwischen Client und Server definiert. Wobei der Grad an Kopplung als Wissen verstanden werden kann, welches der Client über die Nachrichten haben muss. Dieses Wissen wird auch als "Out of Band Knowledge" bezeichnet. Der Verzicht, dieses Wissen innerhalb des Clients zu codieren, sorgt für eine bessere Flexibilität und Wartbarkeit des Gesamtsystems. Hypermediaformate ab der dritten Ebene betten ihre Nachrichten in einen semantischen Kontext. Die Kopplung besteht hier dann ausschließlich aus den definierten Steuerelementen und der Einigung über das Datenformat. Innerhalb dieser Arbeit werden die H-Faktoren und die 4 Design-Elemente als Grundlage verwendet, um verschiedene Formate untereinander vergleichbar zu machen. Diese Ansicht ist nur als Vergleich zu sehen, da Hypermedia Typen nicht zwangsläufig alle H-Faktoren und Design Elemente unterstützen müssen. So kann ein rein zum lesen ausgelegter Medientyp z.B. auf das Design Element des Applikationsflusses verzichten. Ein gängiges Beispiel hierfür ist ein RSS-Feed Reader. Der Medientyp beinhaltet nur eine kurze Vorschau und einen Link auf den gesamten Artikel. Funktionen, die den Applikationsfluss steuern sind nicht notwendig.

### **2.3 Richardsons Maturity Model**

### **2.4 Hypermedia in der Praxis**

### **2.5 Design-Prozess**

### **2.6 Anwendungsbeispiel**

---

<sup>17</sup>Vgl. Amundsen, M. (2010), <http://amundsen.com/hypermedia/scraps/#hypermedia-levels>

## **3 Bewertung**

### **3.1 Bewertungsgrundlage**

## **4 Hypermedia-Formate**

### **4.1 Basisformate**

### **4.2 JSON**

Beschreibung des Formats

#### **4.2.1 Beispielerantworten aus der Anwendung**

#### **4.2.2 Bewertung JSON**

## **5 Fazit und Ausblick**

\*Literaturverzeichnis

Amundsen, M. (2012). *Building Hypermedia APIs with HTML5 and Node*. O'Reilly.

Hesse, W. (1992). *Software-Entwicklung - Vorgehensmodelle, Projektführung, Produktverwaltung Handbuch der Informatik*. Test, Oldenbourg.

## **Ehrenwörtliche Erklärung**

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Bachelorthesis selbstständig und ohne unerlaubte fremde Hilfe angefertigt habe, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Wolfenbüttel, den 14. Mai 2013



