



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# **Bericht INF-M3 SR**

Amine El Ayadi

iLife - intelligentes Leben exemplarisch auf Android

# Amine El Ayadi

## iLife - intelligentes Leben exemplarisch auf Android

Ausarbeitung eingereicht im Rahmen der Veranstaltung Ringvorlesung  
im Studiengang Master Informatik  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Kai von Luck

Abgegeben am 28. Februar 2009

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b> .....	<b>3</b>
<b>1. Einführung und Motivation</b> .....	<b>4</b>
1.1. Zielsetzung und Motivation .....	4
1.2. Gliederung .....	4
<b>2. Definition</b> .....	<b>5</b>
2.1 iCalendar: .....	5
2.2 iBlogger:.....	7
2.3 iOrganizer: .....	8
<b>3. Techniken</b> .....	<b>11</b>
3.1 Platform und Entwicklungs-Umgebung.....	11
3.1.1 Android .....	11
3.1.2 Entwicklung .....	11
3.1.3 Client-Server Architektur .....	12
3.2 Technologies .....	12
3.2.1 RDF (Resource Description Framework) .....	12
3.2.2 Ontologie.....	13
3.2.3 Textmining .....	13
3.2.4 Lucene .....	14
3.2.5 Clusteranalyse .....	14
<b>4. Schluss</b> .....	<b>16</b>
4.1 Ziel / Masterarbeit .....	16
4.2 Risiken .....	16
<b>Quellen</b> .....	<b>17</b>

# **1. Einführung und Motivation**

## **1.1. Zielsetzung und Motivation**

Derzeit verfügt jeder Mensch über mindestens ein Handy. Das Nutzen des Handys beschränkt sich nicht mehr nur auf das Telefonieren, vielmehr hat sich der Focus des mobilen Geräts verändert.

Die Funktionalität erweitert sich auf Kalender-, Kamera-Funktionalitäten und Internetzugang aus. Zudem bietet das Internet immer mehr interessante Dienste, welche über das Handy anwendbar sind und den Benutzer bereichern.

In diesem Zusammenhang können Handy-Anwendungen entwickelt werden, die das Leben besser gestalten bzw. organisieren, in dem ausgewählte Dienste aus dem Handy intelligent genutzt werden.

Google zählt zu den besten Dienstanbietern, die nützliche und kostenlose online Dienste anbieten (z.B. google Calendar, Blogger, Picassa, googleMaps, YouTube), Außerdem bietet Google die Handy Plattform Android, die auf dem Linux-Betriebssystem basiert. Ein großer Teil der Software ist frei und Open Source.

Mit Hilfe von Semantic Web Techniken soll im Rahmen der Masterarbeit die mobile Anwendung iLife entstehen. iLife soll das Leben des Menschen vereinfachen, indem Google Dienste kombiniert werden. Dabei hat iLife den Vorteil, dass ein mobiles Gerät alle benötigten Informationen für die Google-Dienste enthalten kann (z. B: Fotos, Kalender Einträge, SMSs, MMSs usw.)

## **1.2. Gliederung**

Die Arbeit gliedert sich vier Kapitel. Das vorliegende Kapitel 1 gibt einen kurzen Überblick über die Motivation, Zielsetzung und den Aufbau der Arbeit.

Kapitel 2 stellt die Idee und eine vollständige Beschreibung der intelligenten Anwendung iLife vor. In Kapitel 3 werden die Techniken zur Umsetzung dargelegt und im letzten Kapitel 4 folgt eine Bewertung des vorgestellten Vorhabens bzw. die Risiken, die den Erfolg der Anwendung verhindern können.

## 2. Definition

Die Anwendung iLife steht als Abkürzung für „intelligent life“ und enthält drei Hauptkomponenten (iCalendar, iBlogger, iOrganizer). Die drei Komponenten arbeiten zusammen und werden anschließend mit entfernten Applikationen (google Calendar, Blogger, Picassa, google Map und Youtube) synchronisiert. Dabei soll der iCalendar alle Termine, Events, SMSs, MMSs, Fotos und Videos eines Users verwalten (s. Abbildung 2.1), sie anzeigen und synchronisieren, dazu soll der iCalendar die Profile des Handy anhand der eingetragenen Termine aktualisieren. Der iBlogger soll aus dem iCalendar alle Aktivitäten eines Users zusammenstellen und daraus eine Post erzeugen, mit der Möglichkeit, Beschreibungen und Kommentare hinzuzufügen. Der iOrganizer soll durch Web Semantic Verfahren alle ähnlichen Termine von einer Gruppe oder neue übereinstimmende Termine mit anderen Usern finden. Dazu soll iOrganizer dem User die Möglichkeit geben Events intelligent zu publizieren.

Wednesday, October 22, 2008	Thursday, October 23, 2008	Friday, October 24, 2008	Saturday, October 25, 2008
09:00 - 14:00 Vorbereitung tt2	08:15 - 11:30 Praktikum TH1	08:15 - 11:30 Projekt	13:00 - 15:00 Einkaufen
 12:30 	12:00 - 13:30 Vorlesung TH1	 16:00 – 18:00 Fitness	 16:00 – 18:00 Fitness
 13:00 	 13:30 – 14:00 Besprechung	 19:00 Kafee trinken (Sternschanze)	 22:00 Party
 Julia Pliszka 25	 16:00 – 21:00 Grillen		 Leyla 4

Abbildung 2.1: iLife Ansicht

### 2.1 iCalendar:

- **iCalendar -Ansicht:**

Der Benutzer erhält eine Übersicht über einen Kalender. Als erstes kann der Benutzer einen Tages-, Wochen- oder Monats-Ansicht anzeigen lassen. Der Benutzer kann anschließend zu einem beliebigen Tag in der Vergangenheit und Zukunft gehen.

In einem Kalendertag sieht der Benutzer folgende Einträge sortiert nach der Uhrzeit (s. Abbildung 2.2):

- empfangende SMSs, die er an diesem Tag erhalten hat. Diese SMSs werden aus dem Posteingang des mobilen Geräts gezogen.

- empfangende MMSs, die er an diesem Tag erhalten hat. Diese MMSs werden aus dem Posteingang des mobilen Gerätes gezogen.
- Fotos, die er an diesem Tag gemacht hat.. Diese Fotos werden aus dem mobilen Gerät gezogen.
- Videos, die er an diesem Tag aufgenommen hat. Diese Videos werden aus dem mobilen Gerät gezogen.

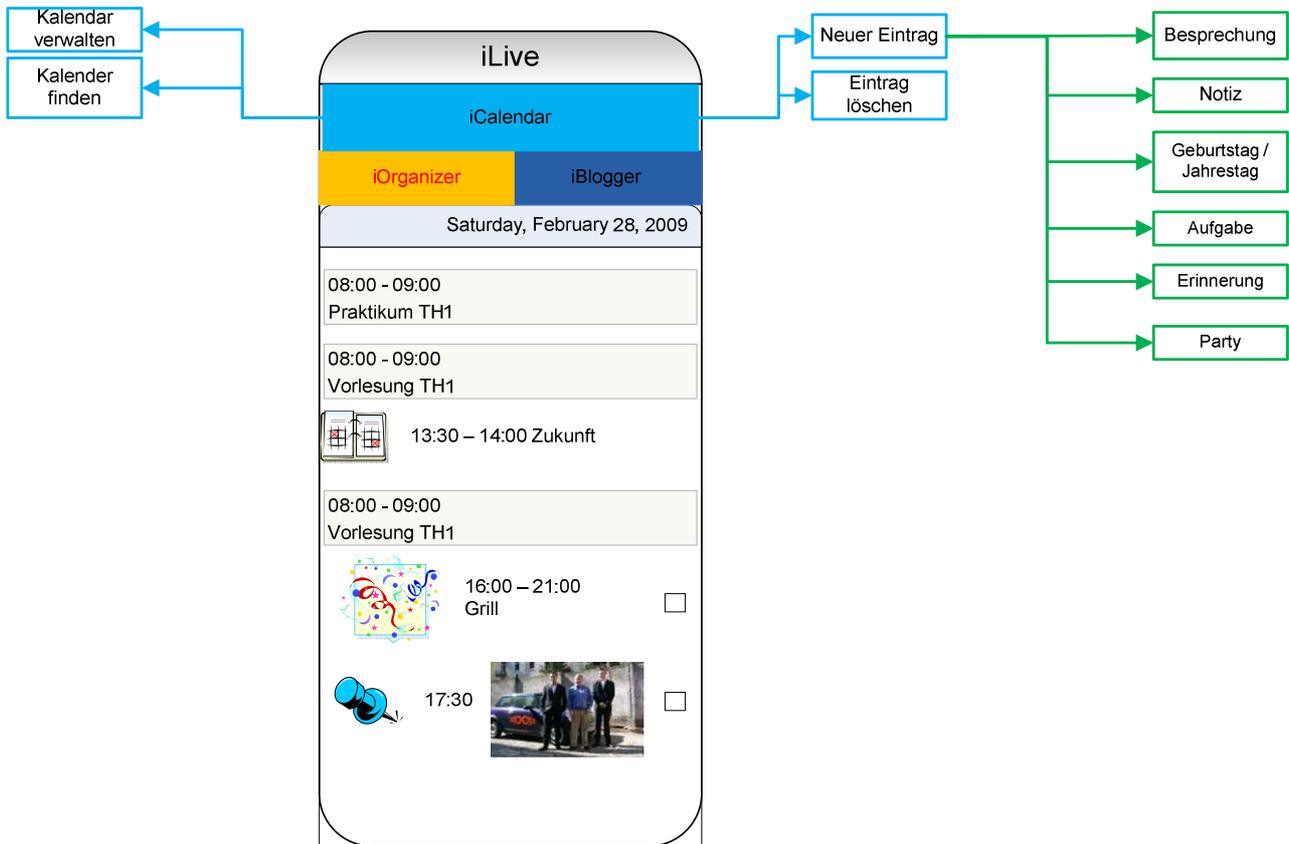


Abbildung 2.2: iLife Tagesansicht

- **iCalendar -Einträge hinzufügen:**

Der Benutzer kann neue Termine erstellen, dabei hat er die Möglichkeit, eins der folgenden Events zu erstellen. Jeder Event ist kategorisiert, und der Benutzer kann für jede Kategorie ein Profil erstellen, welches das Verhalten des Handy beschreibt, z.B. Lautstärke, Klingelton, online/offline usw.

- **Besprechung:** Bei Besprechungen soll der Benutzer einen Betreff, Ort, Datum, Anfangs- und End- Zeit eingeben. Dabei wechselt das Handy automatisch auf das entsprechende Profil.
- **Notiz:** Bei Notizen soll der Benutzer einen Betreff, Ort, Datum, Anfangs- und End- Zeit eingeben.
- **Geburtstag / Jahrestag:** Hier soll der Benutzer den Anlass und das Datum eingeben.

- Aufgabe: Bei Aufgaben soll der Benutzer den Betreff und das Fälligkeitsdatum eingeben, dazu kann der Benutzer eine Erinnerungs-Zeit eingeben.
- Erinnerung: Der Benutzer soll an dieser Stelle einen Betreff und Erinnerungsdatum und –zeit eingeben.
- Party: Bei Parties soll der Benutzer einen Betreff, Ort, Datum, Anfangs- und End- Zeit eingeben. Dabei wechselt das Handy automatisch auf das entsprechende Profil.

- **iCalendar -Einträge in Google-Kalender veröffentlichen:**

Der Benutzer kann optional Termine aus dem iCalendar in seinem Google-Kalender veröffentlichen. Hierfür werden zuerst die Termine gewählt, die veröffentlicht werden sollen, und im Anschluss kann gewählt werden, in welchen Google- Kalender die Einträge geschrieben werden sollen. Der Benutzer hat z.B. zwei Google-Kalender, einen für seine Meeting- und Geschäftsplanung und einen für seine Hochschulplanung. Seine Arbeitskollegen und Vorgesetzten haben Zugriff auf den Meeting- und Geschäftsplanungs-Kalender und seine Kommilitonen auf den Hochschulplanungs-Kalender. Termine wie “mit Kommilitonen etwas trinken gehen“ kann der Benutzer nur dem Hochschulplanungskalender hinzufügen, damit seine Arbeitskollegen und Vorgesetzten keine Einsicht in diesen Termin haben. Weitere Termine, wie z.B. Master-Kolloquium, können dagegen in beiden Kalender veröffentlicht werden.

- **iCalendar -synchronisieren:**

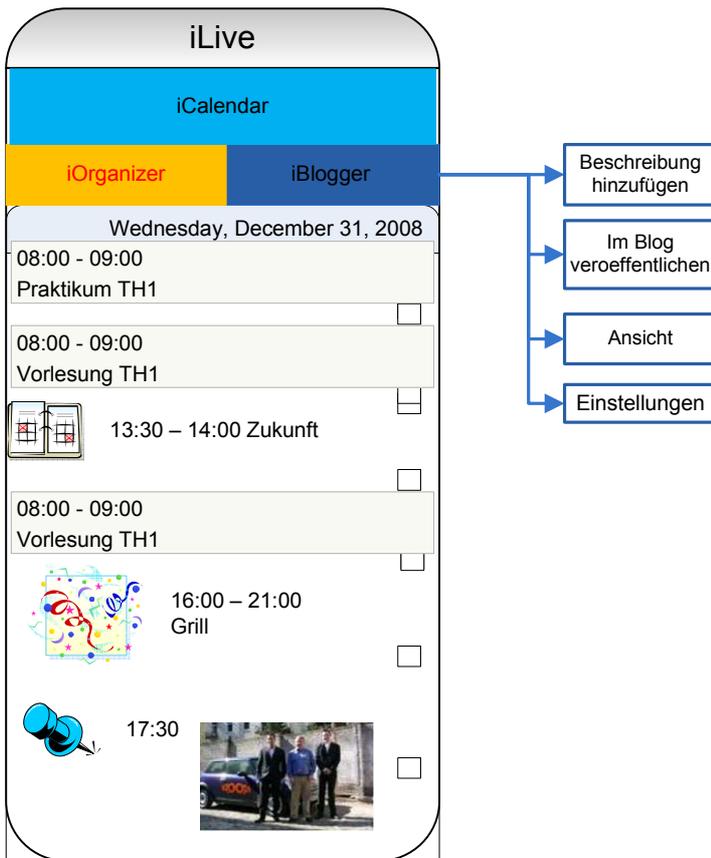
Der Google Kalender wird mit dem Kalender des mobilen Gerätes synchronisiert nach jedem Starten der iCalendar Anwendung. Zusätzlich hat auch der Benutzer die Möglichkeit, über das User Interface die Synchronisation zu starten.

## 2.2 iBlogger:

- **iBlogger -Einträge in einem Blog veröffentlichen:**

Der Benutzer hat die Möglichkeit, die Einträge aus dem iCalendar in einem Blog zu veröffentlichen. Der Benutzer wählt aus dem iCalendar Termine, Fotos, Videos, SMSs und/oder MMSs aus, die er publizieren möchte, dazu können Kommentare und Beschreibungen hinzugefügt werden (s. Abbildung 2.3). Da ein Blog ein Tagebuch darstellt, ist es nur erlaubt, Einträge aus der Vergangenheit im Blog zu publizieren. Diese iCalendar Einträge werden gespeichert und anschließend im google Blogger veröffentlicht. Zudem kann zu jedem Eintrag ein Kommentar vom Benutzer verfasst werden, der im Blog ebenfalls zu sehen ist. Alle im Blog veröffentlichten Fotos werden mit der Ortsbestimmung dargestellt, sodass jeder sehen kann, wo genau die Fotos entstanden sind. Dabei erhält der Benutzer einen Link zu google Maps, um den Ort auf der Karte anzeigen zu lassen. Da die Blogger API die Funktion Foto hochladen nicht unterstützt, werden die Fotos in Picassa hochge-

laden und anschließend in dem Blogger verlinkt. Ebenfalls können Videos über Youtube hochgeladen und ebenfalls in dem Blogger verlinkt werden.



**Abbildung 2.3: Blog Funktionalitäten- Ansicht**

- **iBlogger -Ortsbestimmung beim Foto schießen \ Video Aufnahme:**

Beim schießen eines Fotos oder Aufnahme eines Videos, muss das mobile Gerät den Ort bestimmen. Dieser wird mit dem Foto oder Video abgespeichert.

## 2.3 iOrganizer:

- **iOrganizer -Ähnliche Termine von Gruppen finden:**

Der Benutzer soll die Möglichkeit haben, ähnliche Termine innerhalb von vordefinierten Gruppen oder selektierten Benutzern zu sehen. Es wird nach ähnlichen Terminen gesucht, die am selben Tag und in ähnlichen Zeiträumen stattfinden (s. Abbildung 2.4). Hat z.B. ein Benutzer aus der Gruppe Fußballverein einen Termin am 11.12.2008 im Zeitraum 22:00 – 24:00 Uhr mit der Terminbezeichnung „Kinoabend“ und ein weitere Benutzer derselben Gruppe einen Termin am gleichen Tag mit dem Zeitraum 20:00 – 22:00 Uhr und der Bezeichnung „Ins Kino gehen“, dann soll der Benutzer sehen können, dass diese beiden Benutzer aus derselben Gruppe zwei sehr ähnliche Termine am

gleichen Tag haben. Jetzt kann der Benutzer sich mit den beiden Anderen zu einem Kinobesuch verabreden.

- **iOrganizer -Nach Begriffen in anderen Google-Kalendern suchen:**

Der Benutzer soll nach Suchbegriffen in Google-Kalendern von anderen Benutzern suchen können. Hierzu kann der Benutzer bestimmen, ob er bei nur einem anderen Benutzer nach dem Suchbegriff sucht oder in einer Gruppe. Beide Optionen sind interessant, wenn der Benutzer z.B. sehen möchte, wer von seinen Kommilitonen TT2 lernt, dann sucht er in der gesamten Gruppe. Falls der Benutzer aber z.B. nur wissen möchte, wann eine bestimmte Person wieder kocht, dann ist es sinnvoll, nur bei dem einen Benutzer zu suchen.

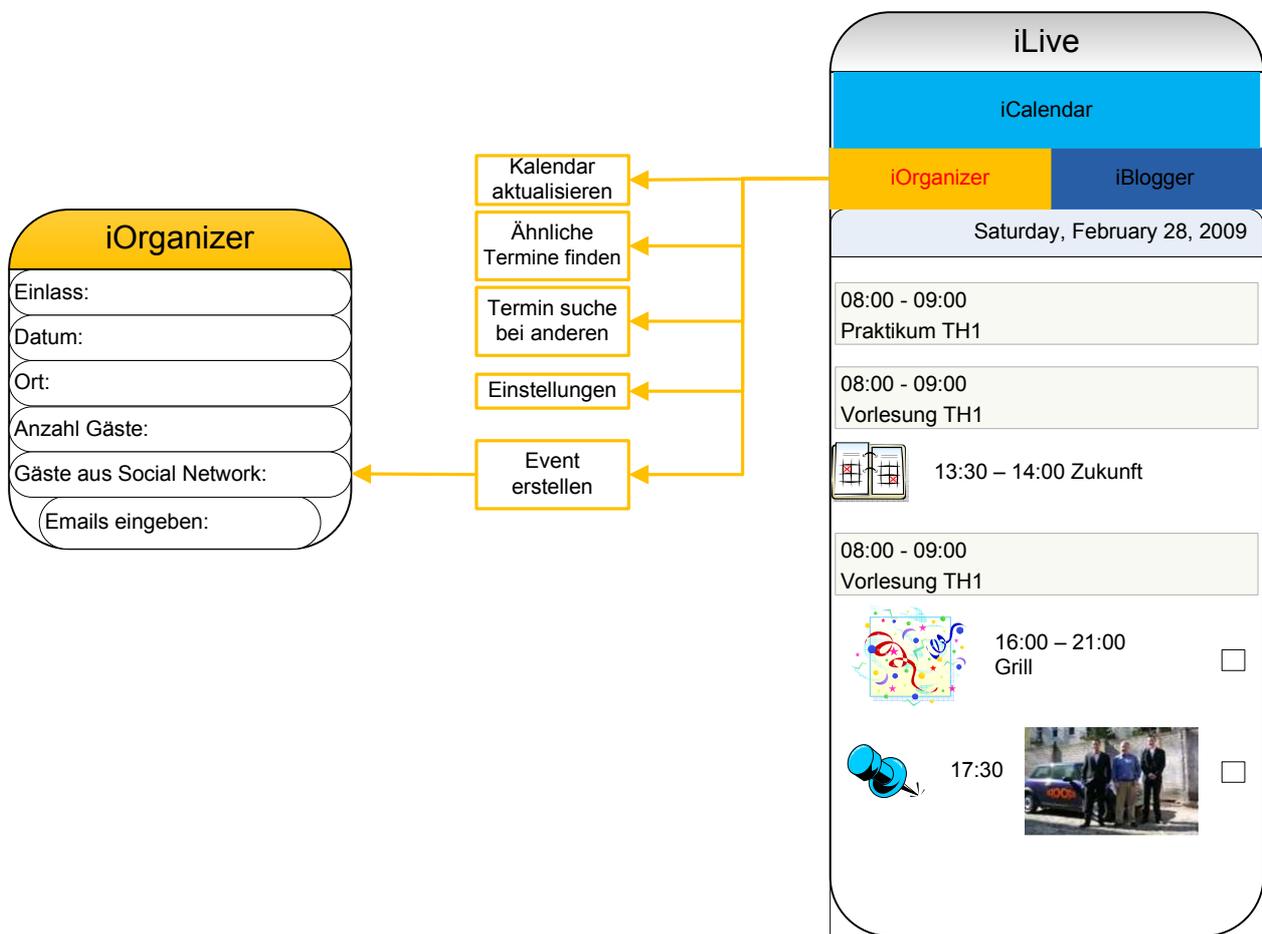


Abbildung 2.4: iLife-iOrganizer

- **iOrganizer -Neue übereinstimmende Termine anzeigen:**

Nach jeder Kalendersynchronisation werden dem Benutzer alle ähnlichen Termine angezeigt, die einer seiner Kontakte mit dem Benutzer gemeinsam haben.

- **iOrganizer -Termine von anderen Benutzern ansehen:**

Der Benutzer soll die Möglichkeit haben, Terminkalender von einem anderen Benutzer anzusehen, damit der Benutzer die Möglichkeit z.B. sehen kann, wann sein Chef wieder geschäftlich verreisen muss.

- **iOrganizer -Event Publisher:**

Der Event-Publisher bietet dem Benutzer, alle seine Events für eine bestimmte Gruppe zu publizieren. Bei dieser Gruppe kann es sich um eine vordefinierte Gruppe oder Benutzer aus der Freundesliste seines Kalenders handeln. Ebenfalls können für die Freunde des Freundes die Events veröffentlicht werden. Es kann bestimmt werden, auf welcher Netzwerk-Ebene die Events veröffentlicht werden sollen, und dabei sollen nur Freunde mit ähnlichen Interessen erkannt und eingeladen werden. Der Benutzer kann ebenfalls allen Kontakten aus seinem Email-Account über seine Events informieren.

## 3. Techniken

### 3.1 Platform und Entwicklungs-Umgebung

#### 3.1.1 Android

Android ist eine vollständige Plattform für mobile Geräte. Sie bringt ein Betriebssystem, eine Laufzeitumgebung, ein Anwendungs-Framework und fertige Kernanwendungen mit. Ist passende Hardware vorhanden, reicht es also, Android zu installieren, um ein Smartphone zu erhalten. Android ist im Umfang etwa vergleichbar mit Windows Mobile, Symbian OS oder der iPhone-Plattform. Es ist allerdings darauf ausgelegt, den Anwendungs-Entwicklern maximalen Nutzen zu bieten: Die Plattform ist quelloffen und leicht erweiterbar. Man kann in Java programmieren. Alle Möglichkeiten der Hardware können genutzt, Applikationen nach Belieben ergänzt oder ersetzt werden (auch Telefonieapplikationen). Zum einen erhält der Entwickler mit Android Freiheiten, die er auf anderen Plattformen nicht hat, und zum anderen erhält er Werkzeuge, mit denen er mobile Anwendungen außergewöhnlich komfortabel entwickeln kann. Beides zusammen soll dazu führen, dass die besten mobilen Anwendungen auf Basis von Android entstehen.

#### 3.1.2 Entwicklung

Anwendungen in Android können mit der SDK entwickelt und mit dem mitgelieferten Emulator getestet werden.

*Android Runtime:* Alle Android-Anwendungen laufen in einem eigenen Prozess in einer Instanz der Android-Laufzeitumgebung. Die Laufzeitumgebung heißt Dalvik Virtual Machine (VM). Das ist eine von Google entwickelte, für mobile Geräte optimierte virtuelle Maschine, die den Android eigenen Bytecode ausführt, den so genannten Dalvik-Bytecode. Wichtig: Die Dalvik VM bringt nicht alle Java APIs einer Java Runtime mit. Die wichtigsten APIs sind aber vorhanden, und man kann eigene JAR-Bibliotheken mit seiner Anwendung mitliefern.

*Anwendungs-Framework:* Das Anwendungs- Framework enthält die grundlegenden Bausteine und APIs, mit denen Android-Anwendungen entwickelt werden. Es ist damit die wichtigste Schnittstelle für alle Android- Anwendungen.

*Anwendungen:* Die Android-Plattform liefert die wichtigsten Anwendungen für ein Smartphone bereits mit: Telefonie, Kontakte, Kalender, Webbrowser, Google Maps und weitere. Alle Anwendungen werden in Java geschrieben und verwenden die Android Runtime. Es ist bisher nicht vorgesehen, dass man native Anwendungen in C/C++ schreibt, die direkt auf Systembibliotheken oder den Linux Kernel zugreifen. Eine echte Besonderheit von Android ist, dass Anwendungen nicht voneinander isoliert sind, sondern über einfache Mechanismen lose miteinander gekoppelt werden können. Entwickler können, wenn die Anwendungen entsprechend gebaut sind, einzelne Funktionalitäten anderer Anwendungen für ihre Anwendung nutzen. Beispiel: Ich möchte in meiner Anwendung die Möglichkeit bieten, einen Kontakt auszuwählen. Anstatt dafür meinen eigenen Dialog zu kreieren, gehe ich einfach in die Kontaktverwaltungsanwendung. Der Benutzer wählt dort den Kontakt aus,

und die aufgerufene Anwendung liefert mir den ausgewählten Kontakt zurück. Auf diese Weise kann ich andere Anwendungen als Dienste betrachten und bei Bedarf in meine Anwendung integrieren.

### 3.1.3 Client-Server Architektur

Um den Umstand der Suche nach Ähnlichen Terminen bzw. Interessen auf das Handy zu vermeiden, soll eine Client Server Architektur entwickelt werden. Die Kommunikation wird mittels Webservices realisiert. Dabei soll der Server zwei Methoden anbieten:

- Ähnliche Termine ermitteln:

Diese Methode erwartet einen Iterator mit allen Terminen und gibt einen Iterator mit den ähnlichen Terminen zurück.

- Profile mit ähnlichen Interessen definieren:

Diese Methode erwartet einen Iterator mit allen Profilen der Freunde auf einem Netzwerk und gibt einen Iterator von den Profilen mit den Profilen, die ähnliche Interessen haben, zurück.

## 3.2 Technologies

Um die ähnliche Termine bzw. Interesse zu erkennen werden erkennen, werden mehrere Techniken bzw. Verfahren von dem Web Semantic benötigt:

### 3.2.1 RDF (Resource Description Framework)

Das Resource Description Framework (RDF) ist eine Sprache, die vom W3C entwickelt wurde, um Metadaten zur Beschreibung von Web-Ressourcen zu erzeugen [LaS99].

Das RDF-Modell ist einfach gehalten und besteht aus drei Objekttypen: Ressourcen, Eigenschaften und Aussagen. Eine Aussage besteht aus einem Tripel (Subjekt, Prädikat, Objekt).

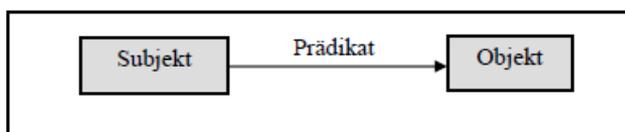
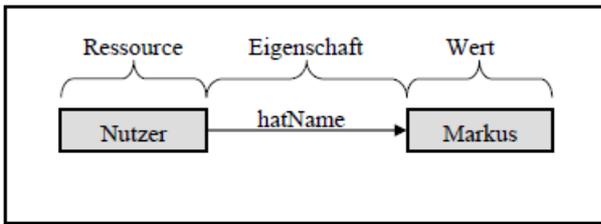


Abbildung 2.2: RDF-Modell

Das Subjekt ist die zu beschreibende Ressource, das Prädikat ist eine Eigenschaft dieser Ressource, und das Objekt dient als ein Wert der Eigenschaft [SGA07]. Hier folgt ein Beispiel:

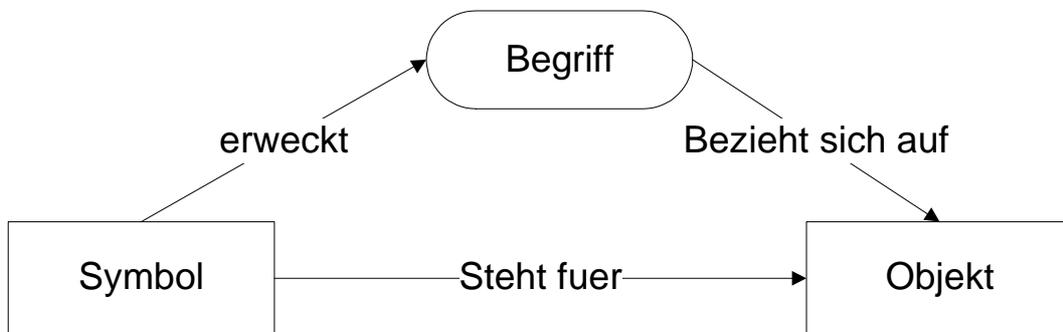


**Abbildung 2.3: RDF-Beispiel**

Für iLife wird RDF verwendet, um einen Termin zu beschreiben bzw. zu kategorisieren, zum Beispiel der User hat einen Termin Fußball der zur Kategorie Sport gehören soll:  
 Ressource ist Fußball, Eigenschaft ist „gehört der Kategorie“ und der Wert ist Sport.

### 3.2.2 Ontologie

Ontologien sind Sammlung und Strukturierung zusammengehöriger Begriffe. Dabei werden Begriffe zueinander in geordnete und hierarchische Beziehung gesetzt:



Die *Ontologies* bieten eine formale spezifizierte Terminologie, die von allen anderen Komponenten verwendet wird [StH05].

### 3.2.3 Textmining

Text Mining bezeichnet die automatisierte Entdeckung relevanter Informationen aus Textdaten. Dabei sollen die relevanten Informationen aus Texten erschlossen werden, indem bedeutungslose Wörter gestrichen werden und Informationen durch Textanalyse gewonnen werden. Dazu soll der Kontext beachtet werden, indem Homonyme differenziert werden und bedeutungsähnliche Begriffe aufeinander gemappt werden [TXTMN 01].

Im Zusammenspiel mit ihren Anwendern sind Werkzeuge und Techniken des Textminings außerdem dazu in der Lage, Hypothesen zu generieren, diese zu überprüfen und schrittweise zu verfeinern. Textmining zählt deshalb auch zu den Verfahren der explorativen Datenanalyse.

Die Herausforderung des Text Mining liegt dabei darin, die in einem Text sprachlich wiedergegebene Information für die maschinelle Analyse zu erschließen.

Es gibt heute zum Teil verlässliche und einfach implementierbare Verfahren, zum Teil ist die zufrieden stellende automatische Lösung dieser Aufgaben Gegenstand der Forschung. Computerlinguistische Verfahren, die diese Aufgaben lösen sollen, benötigen kodiertes Wissen verschiedener Art, z. B. [ACM 01]:

- Wörterbücher, in denen die Wortart(en) und Bedeutung(en) von Wörtern und Phrasen registriert sind,
- Ontologien, die darüber hinaus den spezifischen Wortschatz eines Fachgebiets und die Beziehungen zwischen den Fachwörtern kodieren,
- Listen von Eigennamen, z. B. für Länder, Städte, Firmen,
- Listen von häufig verwendeten Abkürzungen.

Diese Verfahren sollen in der iLife Anwendung verwendet werden, um ähnliche Termine eines Users zu ermitteln. Dabei sollen aus den Terminen die relevanten Informationen und deren Synonyme, Beziehungen und Abkürzungen ermittelt werden. Diese Informationen werden später mit Hilfe von Lucene indiziert, um die Suche nach ähnlichen Terminen zu erleichtern.

### 3.2.4 Lucene

**Lucene** ist eine Open-Source-Java-Bibliothek zum Erzeugen und Durchsuchen von Text-Indizes. Mit ihrer Hilfe lassen sich Volltextsuchen für beliebige Textinhalte implementieren.

Die Bibliothek setzt sich aus zwei Hauptbestandteilen zusammen:

1. Eine Komponente erzeugt den Index, wobei diesem beliebige Textdokumente hinzugefügt werden können.
2. Eine Query Engine durchsucht diesen Index.

Durch die hohe Performanz und Skalierbarkeit kann Lucene für beliebige Projektgrößen und Anforderungen eingesetzt werden.

### 3.2.5 Clusteranalyse

Die Clusteranalyse ist ein Analyseverfahren, das dazu dient, Objekte aufgrund ihrer Ähnlichkeit zu Gruppen zusammenzufassen, wobei die Objekt-Eigenschaften bestimmte Ähnlichkeiten aufweisen sollen.

**Cluster Analyse Verfahren:**

- **Partitionierungsverfahren**

Die Cluster werden so lange optimiert, bis sie extern so heterogen und intern so homogen wie möglich sind.

- **Modell basierte Methoden**

Die Zugehörigkeit zu einem Cluster wird mittels einer Wahrscheinlichkeit unter Annahme einer bestimmten Verteilung (und Anzahl von Clustern) bestimmt.

- **Fuzzy Clustering**

Fordert keine 100% Zugehörigkeit zu einem bestimmten Cluster, sondern erlaubt eine prozentuale Zugehörigkeit

- **Hierarchische Cluster Verfahren**

**Agglomerative:** Die ähnlichsten Cluster werden zusammengefasst.

**Divisive:** Dabei werden die Daten weiter aufgeteilt in heterogenere Cluster.

Das Cluster Verfahren soll für den Event-Publisher in iLife verwendet werden, um Freunde mit ähnlichen Interessen zu identifizieren.

## 4. Schluss

### 4.1 Ziel / Masterarbeit

Die Arbeit beschreibt die Anforderungen der mobilen Anwendung iLife und die möglichen Technologien, die für die Realisierung notwendig sind.

Das Ziel der Masterarbeit ist die Vertiefung der Semantic Web Techniken und die Realisierung einer fertigen mobilen Anwendung, die mit entfernten Applikationen kommuniziert. Diese Kommunikation wurde im Rahmen der AW1 Veranstaltung untersucht, indem die möglichen Technologien „SOAP und REST Web Services“ verglichen wurden. Die möglichen Semantic Web Techniken wurden im Rahmen der AW2 Veranstaltung vorgestellt, eine Vertiefung dieser Technologien ist im Rahmen der Masterarbeit vorgesehen.

Das Konzept für iLife und die notwendigen Basis Applikationen wurden im Rahmen des Projekts **„Pervasive Gaming Framework“** implementiert. Dabei wurde eine Kalender-Anwendung für Android realisiert, dazu wurden die Schnittstellen zu Google-Kalender und zum Social Network „Facebook“ implementiert.

Die Implementierung der Kalender-Anwendung war notwendig, da der Standard Android-Kalender keine Open Source Anwendung ist.

### 4.2 Risiken

Die Masterarbeit stellt komplexe bzw. anspruchsvolle Anforderungen, die eine Vertiefung in mehreren Bereichen erforderlich machen, besonderes, wenn sich einer dieser Bereiche noch in der Forschungsphase befindet. Außerdem ist die vorgegebene Zeitspanne etwas zu kurz, um eine komplette und nutzbare Anwendung zu realisieren, insbesondere, wenn die Android Plattform eine neue Umgebung für mobile Endgeräte sein soll. Durch die Einarbeitung in diese Plattform im Rahmen des Projekts **„Pervasive Gaming Framework“** sind viele Probleme aufgetreten, u.a. die Unkompatibilität der Plattform mit den Google API's, da die Dalvik VM nicht alle Java APIs einer Java Runtime anbietet. Durch diese Risiken kann die vorgegebene Implementierungszeit zu kurz werden und deren erfolgreiche Umsetzung verhindert werden.

# Quellen

- **[LaS99] Resource Description Framework (RDF) Model and Syntax Specification.** W3C Recommendation, 1999.  
*O. Lassila und R. R. Swick.*  
[\[http://www.w3.org/TR/1999/REC-rdf-syntax-19990222\]](http://www.w3.org/TR/1999/REC-rdf-syntax-19990222), Mär. 2008.
  
- **[SGA07] Semantic Web Services: Concepts, Technologies and Applications.** Springer, 2007.  
*R. Studer, S. Grimm, A. Abecker.*
  
- **[StH05] Semantic Web Services Tutorial.** DERI-Digital Enterprise Research Institute. 3rd International Conference on Web Services, 2005.  
*M. Stollberg und A. Haller.*
  
- **[TXTMN 01] Text Mining: Wissensgewinnung aus natürlichsprachigen Dokumenten.** *René Witte und Jutta Mülle*  
[\[http://www.rene-witte.net/text-mining-report-2006\]](http://www.rene-witte.net/text-mining-report-2006)
  
- **[ACM 01] Differentiating data- and text-mining terminology**  
*Jan H. Kroeze, Machdel C. Matthee, Theo J. D. Bothma* (Department of Informatics, School of IT, University of Pretoria, Pretoria, 0002)  
[\[http://portal.acm.org/ft\\_gateway.cfm?id=954024&type=pdf&coll=Portal&dl=ACM&CFID=23776836&CFTOKEN=57582297\]](http://portal.acm.org/ft_gateway.cfm?id=954024&type=pdf&coll=Portal&dl=ACM&CFID=23776836&CFTOKEN=57582297)