

**Kontextsensitive
Informationsstrukturierung und -auswahl
für standortbezogenes Mobile Learning
unter Zuhilfenahme von Ontologien**

ALEXANDER PICHLER

MASTERARBEIT

eingereicht am
Fachhochschul-Masterstudiengang

INTERACTIVE MEDIA

in Hagenberg

im Oktober 2012

© Copyright 2012 Alexander Pichler

Diese Arbeit wird unter den Bedingungen der *Creative Commons Lizenz Namensnennung–NichtKommerziell–KeineBearbeitung Österreich* (CC BY-NC-ND) veröffentlicht – siehe <http://creativecommons.org/licenses/by-nc-nd/3.0/at/>.

Erklärung

Ich erkläre eidesstattlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen entnommenen Stellen als solche gekennzeichnet habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Hagenberg, am 4. Oktober 2012

Alexander Pichler

Inhaltsverzeichnis

Erklärung	iii
Kurzfassung	vii
Abstract	viii
1 Einleitung	1
1.1 Problemstellung	1
1.2 Motivation	2
1.3 Zielsetzung	2
1.4 Gliederung der Arbeit	3
2 Grundlagen des mobilen Lernens	4
2.1 Abgrenzung des Begriffs	4
2.1.1 Definition	4
2.1.2 Mögliche Endgeräte	6
2.2 Lerntheoretische Rahmenbedingungen	6
2.2.1 Individuelles Lernen	8
2.2.2 Situiertes Lernen	8
2.2.3 Kollaboratives Lernen	9
2.2.4 Ergänzung zu Ryus und Parsons Kategorien	9
2.3 Potenziale	10
2.3.1 Technologischer Fortschritt	10
2.3.2 Hoher Verbreitungsgrad	12
2.3.3 Zeit- und Ortsunabhängigkeit	12
2.3.4 Lernen im authentischen Kontext	13
2.4 Einschränkungen und Eigenheiten	13
2.4.1 Technologische Barrieren	14
2.4.2 Geminderte Konzentrationsfähigkeit	14
2.4.3 Aufbereitung der Lernmaterialien	15
2.5 Bedeutsamkeit und Definition der Kontextinformationen	15
2.6 Relevante Kategorien von Kontextinformationen	16
3 Informationsstrukturierung und -auswahl	20

3.1	Grundlagen Ontologien	20
3.1.1	Definition	21
3.1.2	Web Ontology Language	21
3.1.3	SPARQL	25
3.2	Alternativen	27
4	Verwandte Arbeiten	31
4.1	m-LOCO: An Ontology-based Framework for Context-Aware Mobile Learning	31
4.2	Ontology-based Framework for Context-Aware Mobile Learning	33
4.3	Context-Aware Mobile Learning with a Semantic Service-Ori- ented Infrastructure	33
5	Eigener Ansatz	36
5.1	Anforderungen	36
5.2	Interaktionsmöglichkeiten	38
5.3	Aufbereitung der Entitäten und Kontextinformationen	39
5.3.1	Zeit	40
5.3.2	Standort	42
5.3.3	Lernende	42
5.3.4	Lernobjekte	45
5.4	Auswahl der Lernobjekte	48
5.4.1	Vorauswahl	48
5.4.2	Reihung	51
6	Umsetzung	53
6.1	Server-Architektur	54
6.1.1	Sesame-Framework	54
6.1.2	Aufbau	57
6.1.3	Schnittstellen für den Client	59
6.2	Client-Architektur	60
6.2.1	Zielplattform	60
6.2.2	Aufbau	60
6.2.3	Benutzerschnittstellen	62
7	Evaluierung	66
7.1	Fallstudie	66
7.1.1	Szenario	66
7.1.2	Befüllen der Wissensbasis	67
7.1.3	Client-Ablauf	68
7.2	Vergleich mit theoretischer Ausgangslage	70
7.2.1	Einhaltung der Grundlagen von Mobile Learning	71
7.2.2	Verwertung von Potenzialen	73
7.2.3	Beachtung der Einschränkungen	74

Inhaltsverzeichnis	vi
7.3 Fazit	74
8 Schlussbemerkungen	76
8.1 Zusammenfassung	76
8.2 Resümee	77
8.3 Ausblick	78
A Inhalt der CD-ROM	79
A.1 PDF-Dateien	79
A.2 Online-Quellen	79
A.3 Projektdateien	79
Quellenverzeichnis	80
Literatur	80
Online-Quellen	83

Kurzfassung

Mobile Learning bedient sich aktueller Technologien, um strukturiertes Lernen an beliebigen Orten zu ermöglichen. Dadurch ergeben sich vor allem in Unternehmen und im tertiären Bildungsbereich zahlreiche Einsatzgebiete. Insbesondere die hohe Wichtigkeit von situiertem sowie individuellem Lernen zeichnen mobiles Lernen aus, was bedeutet, dass der Benutzer, dessen Präferenzen und der Kontext, in dem diese Lernerfahrung erfolgt, berücksichtigt werden müssen. Gleichzeitig sind im Zusammenhang mit Entwicklungen des semantischen Webs neue Möglichkeiten entstanden, die Repräsentation von Wissen zu erleichtern, dieses maschinenlesbar zu machen und daraus automatisiert neue Erkenntnisse abzuleiten.

Eine Kombination dieser Ansätze scheint vielversprechend. Aus diesem Grund wird im Rahmen der vorliegenden Arbeit ein Mobile Learning Framework vorgeschlagen, das Kontextinformationen ontologiebasiert verwaltet. Dadurch kann die Selektion der Lerninhalte systemseitig unterstützt werden, weshalb diese Auswahl zum richtigen Zeitpunkt, automatisch und basierend auf vorhandenen Kontextinformationen getroffen werden kann. Die Entwicklung dieses Systems erfolgt basierend auf Grundlagen des mobilen Lernens, wobei dem ortsbezogenen Kontext ein hoher Stellenwert eingeräumt wird. Insbesondere der Strukturierung von Kontextinformationen, einer darauf basierenden Auswahl von Lerninhalten und der Gestaltung des Clients kommt eine hohe Bedeutung zu.

Abstract

Mobile learning uses the latest technologies to provide structured learning experiences at any location. This results in numerous fields of application, especially in companies and higher education. Concepts such as situated learning as well as individual learning are of high importance. This means that not only the users themselves but also their preferences and the context in which learning experiences take place must be taken into account. At the same time opportunities related to the Semantic Web have arisen which facilitate the representation of knowledge, make it machine-readable and enable computers to automatically infer new knowledge.

Combinations of mobile learning and Semantic Web technologies seem promising. For that reason, a mobile learning framework is being proposed which allows ontology-based management of information. This enables the system to automatically select learning content at the right time and based on available context information. The development of this framework is based on theoretical foundations of mobile learning with emphasis on location-related context. In particular, the structuring of context information, the selection of learning objects as well as the client design will be presented in detail.

Kapitel 1

Einleitung

1.1 Problemstellung

Mobile Learning hat bereits einen wichtigen Stellenwert und gewinnt nach wie vor an Bedeutung, was sich vor allem aus der höheren Verbreitung und technischen Verbesserungen mobiler Endgeräte ergibt. Individuelles sowie situierendes Lernen sind hierbei von hoher Wichtigkeit, was bedeutet, dass der Benutzer und dessen Präferenzen im Mittelpunkt stehen und in enger Verbindung mit dem Kontext, in welchem diese Lernerfahrung erfolgt, gesehen werden müssen. Diese Kontextinformationen zu modellieren stellt eine besondere Herausforderung dar.

Demgegenüber stehen Entwicklungen im Bereich des Semantischen Webs, welche die Repräsentation von Wissen erleichtern, dieses maschinenlesbar machen und daraus automatisiert neue Erkenntnisse ableiten lassen. Um komplexe Zusammenhänge beschreiben zu können, bieten sich insbesondere Ontologien mit ihrer Ausdrucksstärke, Flexibilität und Erweiterbarkeit als Lösung an. Für mobiles Lernen ist es notwendig, Lernobjekte sowie Kontextinformationen in einem durchdachten System gleichermaßen effektiv zu verwalten, weshalb sich eine Umsetzung unter Zuhilfenahme von Werkzeugen des semantischen Webs anbietet.

Es soll eine für zahlreiche Anwendungsfälle wiederverwendbare Lösung gefunden werden, welche Lernobjekte durch das System automatisch kontextabhängig selektiert und den Benutzern zur Verfügung stellt. Dennoch sollten Anwendern Möglichkeiten zur eigenständigen Auswahl geboten werden, um benutzergesteuerte Exploration der Lerninhalte zu ermöglichen. Dafür ist es wichtig, dass der Client, welcher die primäre Schnittstelle zum Anwender darstellt, alle systemseitig vorhandenen Funktionalitäten zugänglich macht, wobei unnötige Komplexität vermieden werden sollte. Zusätzlich zeichnet sich Mobile Learning vor allem durch die Mobilität der Lernenden aus, weshalb das Einsatzgebiet des geplanten Frameworks primär auf ortsabhängige mobile Lernanwendungen gelegt wird.

1.2 Motivation

Mobiles Lernen wie auch Technologien des semantischen Webs existieren zwar schon seit einigen Jahren, erreichen mittlerweile aber Reifegrade, die sinnvolle, effiziente und einsetzbare Lösungen erlauben. Dies begründet sich unter anderem darauf, dass das Verständnis von Mobile Learning in der Literatur zunehmend einheitlicher wird und Semantic Web Frameworks die Entwicklung von umfangreichen Applikationen unterstützen. Diese Ausgangslage erlaubt die Umsetzung eines Mobile Learning Frameworks, welches auf umfangreichen theoretischen Erkenntnissen basiert und durch den Einsatz aktuellster Technologien möglich wird.

Um für das geplante System ein praktisches Anwendungsgebiet zu schaffen, wird als Szenario die Umsetzung einer Applikation zum Kennenlernen von Orten angenommen. Derzeitige Umsetzungen von mobilen Applikationen zu diesem Zweck beschränken sich vor allem auf touristische Einsatzgebiete und erfordern anwenderseitigen Aufwand bei der Auswahl gewünschter Informationen. Die systemseitige Unterstützung ist dabei meist auf wenige Mittel, wie eine Umkreissuche in Kombination mit einer Filterung nach Kategorien, beschränkt. Dadurch ist es beispielsweise möglich, gastronomische Einrichtungen in nächster Nähe zum Aufenthaltsort anzuzeigen. Eine mobile Lernanwendung zum geleiteten Kennenlernen von Orten, welche eine größere Anzahl an Kontextinformationen integriert und theoretische Grundlagen von Mobile Learning berücksichtigt, existiert in dieser Form noch nicht.

1.3 Zielsetzung

Das Ziel dieser Arbeit ist die Erstellung eines kompakten Frameworks, welches mobiles Lernen im Hinblick auf die theoretischen Grundlagen umsetzt. Dabei wird vor allem der kontextabhängigen Auswahl von Inhalten ein hoher Stellenwert eingeräumt, was durch Methoden des semantischen Webs ermöglicht werden soll. Die Umsetzungsmöglichkeit eines solchen Ansatzes wird überprüft und anhand einer Fallstudie praktisch evaluiert, wobei als Anwendungsgebiet das vom System geleitete Kennenlernen eines Ortes angenommen wird.

Als zentrale Fragestellung gilt es zu klären, wie Informationen für standortbezogene mobile Lernanwendungen ontologiebasiert strukturiert und daraufhin kontextabhängig selektiert werden können. Dabei soll vor allem darauf eingegangen werden, welche Kontextinformationen durch die Ontologie repräsentiert werden können, wie diese Repräsentation aufgebaut sein kann und wie in weiterer Folge vorhandene Lerninhalte kontextabhängig ausgewählt und an den Benutzer übermittelt werden können. Darüber hinaus ist zu klären, wie der mobile Client gestaltet werden soll, um den Benutzer bei der Verwendung des Systems zu unterstützen. Hinsichtlich dieser Ziele wird

der Einhaltung von Grundlagen des mobilen Lernens ein hoher Stellenwert eingeräumt.

1.4 Gliederung der Arbeit

Im Rahmen einer theoretischen Abklärung des Themas Mobile Learning wird in Kapitel 2 auf dessen Eigenheiten und Potentiale eingegangen. Ebenfalls werden einführende Informationen zum Thema Kontext präsentiert. Auf diese Weise wird dargestellt, warum es nötig ist, sich mit einer ontologiebasierten Modellierung von Kontextinformationen und einer darauf basierenden Lernobjektauswahl auseinanderzusetzen. Im darauffolgenden Kapitel werden mögliche Strukturierungsmöglichkeiten für diese Informationen beschrieben, mit einem Schwerpunkt auf Ontologien und darauf bezogene technische Möglichkeiten. Das letzte, dem überwiegend theoretischen Bereich zugeordnete Kapitel widmet sich verwandten Arbeiten, welche sich ebenfalls mit ontologiebasierten mobilen Lernanwendungen beschäftigen.

Basierend auf diesen Grundlagen wird in Kapitel 5 der eigene Ansatz besprochen, wobei die verwalteten Kontextinformationen und der Lernobjekt-Auswahlprozess dargestellt werden. Ebenfalls ist Kapitel 6 durch die genauere Beschreibung der Server- und Clientarchitektur praktischen Inhalten gewidmet, wobei eine Erläuterung der Benutzerschnittstellen enthalten ist. In Kapitel 7 erfolgt eine Evaluierung der Lösung anhand der Präsentation einer Fallstudie und eines Vergleiches mit den theoretischen Grundlagen. Die wichtigsten Erkenntnisse werden im letzten Kapitel zusammengefasst, das darüber hinaus versucht, einen kurzen Ausblick zu geben.

Kapitel 2

Grundlagen des mobilen Lernens

2.1 Abgrenzung des Begriffs

Mobile Learning hat in den letzten Jahren an Bedeutung gewonnen, wobei technische Weiterentwicklungen und ein größerer Bekanntheitsgrad der Thematik auf ein Anhalten dieser Tendenz hinweisen. Dennoch ist in aktueller Literatur keine einheitliche Definition des Begriffs zu finden. Es bestehen jedoch Gemeinsamkeiten zum mittlerweile weit verbreiteten E-Learning, welches durch Kohn [13, S. 4] wie folgt definiert wird:

E-Learning (= elektronisches Lernen) umfasst alle Möglichkeiten und Varianten des Lernens mit elektronischen Medien. Der E-Learning-Begriff im weitesten Sinn schließt also sowohl Online-Medien (WBT, Streaming, Videos, etc.) als auch Offline-Medien und -Methoden (CBT, DVD, CD-ROM) ein.

Lernende können somit Zeitpunkt und Lernort zu einem überwiegenden Teil selbst bestimmen, sind bei der Wahl des Lernortes jedoch an die Verfügbarkeit eines Computers gebunden. Diese Einschränkung lässt sich bei längeren, geplanten Lernphasen vermutlich umgehen, stellt jedoch eine zu beachtende Minderung des Freiheitsgrades dar. Diese Restriktion ist beim eigenständigen Konzept Mobile Learning durch die Verwendung von transportablen Geräten nicht gegeben [21].

Der Begriff Mobile Learning umfasst aber noch weitere Eigenschaften, welche im Folgenden näher abgegrenzt werden.

2.1.1 Definition

Eine weit verbreitete Interpretation des Begriffs haben O'Malley et al. [35, S. 6] in einer Veröffentlichung geschaffen, welche unter anderem eine genauere Definition von Mobile Learning zum Ziel hatte:

Any sort of learning that happens when the learner is not at a fixed, predetermined location, or learning that happens when the learner takes advantage of the learning opportunities offered by mobile technologies.

Diese Formulierung lässt bereits erkennen, dass Mobile Learning häufig aus zwei Perspektiven betrachtet wird. Einerseits spielt die Verwendung mobiler Technologien aus verständlichen Gründen eine wichtige Rolle, andererseits hat sich die Mobilität aus Sicht der Lernenden vor allem in aktuelleren Publikationen als maßgeblichere Betrachtungsweise durchgesetzt. Dies wird etwa durch Pachler et al. [22, S. 6] verdeutlicht:

Definitions of mobile learning in the literature are manifold but they tend to revolve around the mobility of the technology or the mobility of the learner with a clear change in emphasis of late from the former to the latter.

Das einfache Übermitteln von Inhalten an mobile Geräte reizt damit nicht die Möglichkeiten mobilen Lernens aus. Stattdessen geht es vielmehr um Prozesse, welche Informationen über die sich laufend verändernden Kontexte¹ und Lernräume generieren und diese in weiterer Folge, vor allem beim Lernen in alltäglichen Situationen, sinnvoll nutzen. Mobile Learning unterstützt den Lernenden beim Wissensaufbau, wobei die Geräte vorwiegend eine vermittelnde Rolle spielen [22].

An dieser Stelle ist erwähnenswert, dass sich die Auffassung von Lernen und Technologie einer vergleichbaren Entwicklung unterzogen hat, wie in Tabelle 2.1 ersichtlich ist:

<i>Neues Lernen</i>	<i>Neue Technologie</i>
Personalisiert	Persönlich
Lernerorientiert	Benutzerorientiert
Situiert	Mobil
Kollaborativ	Vernetzt
Allgegenwärtig	Allgegenwärtig
Lebenslang	Langlebig

Tabelle 2.1: Konvergenz zwischen Lernen und Technologie [25].

Ebenso wie sich das Verständnis von Lernen so veränderte, dass der Lernende in den Mittelpunkt gerückt ist, wurden auch personalisierte digitale Technologien und Services umgesetzt, wie beispielsweise elektronische Kalender. Auch die Sichtweise, dass es sich beim Lernvorgang um eine situierte

¹Eine genauere Definition des Begriffs Kontext kann in Abschnitt 2.5 gefunden werden.

und kollaborative Aktivität handelt, wird durch technologische Entwicklungen widergespiegelt. Anwendern ist es möglich, sich unabhängig von ihrem Standort zu vernetzen, Wissen auszutauschen oder gemeinsam Probleme zu lösen. Ob beruflich oder privat, Menschen lernen ein Leben lang und werden dabei häufig von verschiedensten Technologien begleitet und unterstützt [25].

2.1.2 Mögliche Endgeräte

Eine weitere zu klärende Frage ist, mit welchen Geräten Mobile Learning betrieben werden kann. An dieser Stelle müssen vor allem Smartphones und Personal Digital Assistants (PDAs) erwähnt werden, welche den größten Funktionsumfang in Verbindung mit hoher Mobilität bieten. Einfache Mobiltelefone werden primär für Kommunikationszwecke eingesetzt, wie beispielsweise Sprachkommunikation, SMS² oder MMS³ und sind im Vergleich zu Smartphones für weniger Verwendungszwecke geeignet.

PDAs stellen kleine tragbare Computer dar, die insbesondere als persönliches Organisationstool verwendet werden und im Normalfall nicht direkt an das Mobilfunknetz angeschlossen sind. Aus diesem Grund spielen sie am Markt eine untergeordnete Rolle und werden immer mehr von Smartphones ersetzt. Diese stellen eine Zusammenführung von Mobiltelefonen mit PDAs dar und erlauben es dem Benutzer, zusätzliche Programme, auch von anderen Anbietern, zu installieren. Aufgrund der Verbreitung und des hohen technischen Funktionsumfangs sind Smartphones ideal für mobiles Lernen qualifiziert.

Laptops und Tablet-PCs sind im Hinblick auf ihre Verwendbarkeit für Mobile Learning kritisch zu bewerten, weil sie aufgrund ihrer Größe seltener mitgenommen werden [30]. Zusätzlich können sie nur zum Teil eine Verbindung mit dem Mobilfunknetz aufbauen. Der Vollständigkeit halber werden an dieser Stelle noch E-Book-Reader, Handheld-Konsolen und Portable-Media-Player erwähnt, welche jedoch für sehr spezielle Verwendungszwecke konzipiert wurden und deshalb nur unter Vorbehalt als geeignete Endgeräte angeführt werden können.

2.2 Lerntheoretische Rahmenbedingungen

Um Mobile Learning aus lerntheoretischer Sicht zu beschreiben, werden an dieser Stelle klassische Theorien in Beziehung mit den Eigenheiten, die mobiles Lernen innehat, gesetzt. Eine mögliche Gruppierung wird von Ryu und Parsons [23] beschrieben, welche sie als *three pillars of learning* bezeichnen.

²Short Message Service (SMS) ist ein textbasierter Dienst zum Übertragen von kurzen Nachrichten.

³Multimedia Message Service (MMS) stellt eine Erweiterung zu SMS dar und ermöglicht eine Übertragung von multimedialen Inhalten wie Fotos, Videos oder Audios.

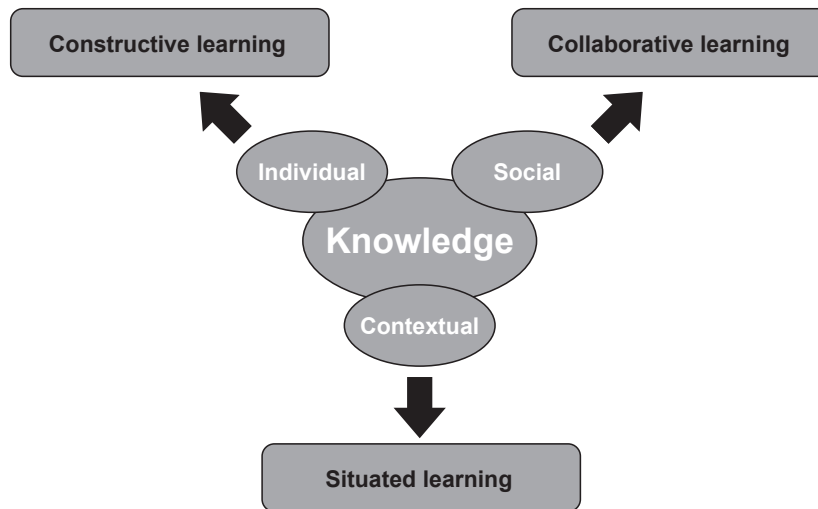


Abbildung 2.1: Drei Lernräume [11].

Diese umfassen situiertes, individuelles und kollaboratives Lernen und basieren auf einer Grafik von De Jong, die in Abbildung 2.1 dargestellt ist.

De Jong [11, S. 431] erläutert die neuen Sichtweisen auf das Lernen im Zusammenhang mit dieser Illustration wie folgt:

More specifically, we can say that the new view on learning entails that learners are encouraged to construct their own knowledge (instead of copying it from an authority be it a book or a teacher), in realistic situations (instead of merely decontextualised, formal situations such as the classroom), together with others (instead of on their own).

Lernende werden heutzutage als Individuen betrachtet, die ihr Wissen nicht einfach aus vorgegebenen Unterlagen übernehmen. Sie erarbeiten sich neue Erkenntnisse selbst, setzen diese in Verbindung mit bestehendem Wissen und weisen dementsprechend Eigenheiten auf, wie sie etwas lernen. Diese Art des Lernens sollte vornehmlich in lebensnahen Situationen passieren, welche das Gelernte anwenden lassen und praktische Zusammenhänge verdeutlichen. Dabei können die Lernenden Erkenntnisse im Diskurs mit anderen vertiefen und erweitern, wobei sie auch die Möglichkeit bekommen, ihr Wissen weiterzugeben. Ryu und Parsons [23] haben diese Kategorien aufgegriffen und individuelles, situiertes sowie kollaboratives Lernen in Verbindung mit Mobile Learning gebracht.

2.2.1 Individuelles Lernen

Lernende werden beim individuellen Lernen als zentraler Punkt gesehen, mit all ihren Erwartungen und vorhandenen Erfahrungen. Diese Ansicht basiert vorwiegend auf Annahmen aus dem Konstruktivismus, der eine zentrale Theorie der Lernpsychologie darstellt und durch Bednar et al. [2, S. 21] wie folgt definiert wird:

[...] the learner is building an internal representation of knowledge, a personal interpretation of experience. This representation is constantly open to change, its structure and linkages forming the foundation to which other knowledge structures are appended. Learning is an active process in which meaning is developed on the basis of experience.

Diese Formulierung besagt, dass jeder Lernende eine interne Wissensrepräsentation besitzt, welche bisher gemachte persönliche Erfahrungen in sich vereint. Neue Erfahrungen interagieren mit ihr und verändern deren Struktur und Verknüpfungen, wodurch eine aktualisierte Ausgangsbasis für weitere Erlebnisse entsteht. Wissen stellt eine Interpretation der erlebten Wirklichkeit dar, die individuell für jeden Lernenden ausgeprägt ist.

Bezugnehmend auf dieses Lernparadigma beschreiben Ryu und Parsons [23] den Lernprozess so, dass Lernende Wissen durch Testen von neuen Ideen und Ansätzen konstruieren, wobei diese auf bestehendem Vorwissen und Erfahrungen basieren. Letztere werden aktiv in neuen Situationen angewendet, wodurch so entstehende Erkenntnisse in bestehendes Wissen integriert werden. Die Aufgabe der Lehrenden besteht nun vielmehr darin, eine angemessene Lernumgebung anzubieten und die Lernenden bei ihrem persönlichen Wissensaufbau zu unterstützen, als vorgefertigte Lernmaterialien in einem zuvor festgelegten, sequentiellen Schema zu übermitteln.

2.2.2 Situiertes Lernen

Bei situiertem Lernen wird der Lernvorgang nicht als eigenständiger, unabhängiger Prozess betrachtet – er wird vielmehr eingebettet in dem Kontext, in welchem er stattfindet, gesehen. Ryu und Parsons [23, S. 21] weisen, bezugnehmend auf eine Publikation von Lave und Wenger [16], auf den hohen Stellenwert einer authentischen Lernumgebung hin:

Lave and Wenger's (1991) situated learning theory differentiates this situated learning space from the other learning spaces, acknowledging that learning needs to be presented in an authentic context, that is, settings and applications that would normally involve that knowledge.

Diese Aussage lässt darauf schließen, dass differenziert werden muss, ob eine Lernerfahrung in einem dekontextualisierten Fortbildungsraum erfolgt, oder in realistischen Situationen, welche erworbenes Wissen womöglich direkt anwenden und festigen lassen. Darüber hinaus ermöglichen Informationen über den Kontext, in dem sich die Lernenden befinden, Lernmaterialien anzupassen und auf die jeweilige Person sowie deren aktuelle Situation abzustimmen. Lave und Wenger [16] bringen situiertes Lernen auch stark mit dem sozialen Umfeld der Lernenden in Verbindung.

2.2.3 Kollaboratives Lernen

Lernen in Gruppen hat gegenüber anderen Methoden den Vorteil, dass mehr von den vermittelten Inhalten über einen längeren Zeitraum behalten wird. Der Austausch zwischen den Lernenden über ihre individuellen Auffassungen fördert damit die Effektivität des Lernvorganges und es wird nicht nur von vorgegebenen Materialien gelernt, sondern auch voneinander [23].

Ein besonderer Ansatz in diesem Bereich wird als computerunterstütztes kollaboratives Lernen bezeichnet. Diese Herangehensweise macht sich Informations- und Kommunikationstechnologien zunutze, wodurch kollaboratives Lernen nicht nur dann möglich ist, wenn die Lernenden zur gleichen Zeit am selben Ort sind. Geräte, die für Mobile Learning geeignet sind, werden unter anderem durch ihre Verwendbarkeit als mobiles Kommunikationsmittel charakterisiert. Dies ergibt eine wesentliche Eignung für Lernsituationen, in denen kollaboratives Lernen eingesetzt werden soll.

2.2.4 Ergänzung zu Ryus und Parsons Kategorien

Naismith et al. [20] gehen neben den bereits dargelegten Lernkategorien auf weitere Punkte ein:

Informelles, lebenslanges Lernen

In diesem Abschnitt sind zwei Begriffe vereint, die starken Bezug zueinander aufweisen. Lebenslanges Lernen spiegelt die Notwendigkeit der modernen Wissensgesellschaft wider, Kenntnisse und Fähigkeiten aktuell zu halten, um vor allem im Berufsleben konkurrenzfähig zu bleiben. Informelles Lernen wird laut Dohmen [8, S. 25] folgendermaßen definiert:

[...] der Begriff des informellen Lernens wird auf alles Selbstlernen bezogen, das sich in unmittelbaren Lebens- und Erfahrungszusammenhängen außerhalb des formalen Bildungswesens entwickelt.

Diese Art des Lernens findet in alltäglichen Situationen statt, beispielsweise bei Konversationen, beim Betrachten der Abendnachrichten oder durch

das Überwinden von schwierigen Umständen, wobei meist kein konkretes Lernziel vorgegeben und der Vorgang nicht speziell strukturiert ist.

Mobile Technologien sind meist fix in den Alltag eingebunden, weshalb sie hervorragend dafür geeignet sind, Menschen bei dieser Art der Wissensgenerierung (beispielsweise durch allorts verfügbaren Zugriff auf zusätzliche Wissensbasen) zu unterstützen. Dies geschieht übergangslos und unaufdringlich. Insbesondere bei unvorhersehbarem, zufälligen Lernen ermöglichen mobile Technologien Dokumentation, Reflexion und Weitergabe von derartigen informellen Lernepisoden [20].

Unterstützung beim Lehren und Lernen

Bisher wurde vor allem auf Möglichkeiten des mobilen Lernens im Zusammenhang mit den Lernenden eingegangen, doch auch für Lehrende ergeben sich Einsatzgebiete. Der Zugang zu den Lernenden kann unmittelbarer gestaltet werden, was bedeutet, dass Kommunikation sowie Feedback kurzfristiger und unabhängig vom Aufenthaltsort stattfinden kann.

Laut Froberg [9] ergeben sich in Unterrichtsräumen über sogenannte Classroom Response Systeme (CRS) weitere Anwendungsmöglichkeiten, durch welche die Lehrenden über Fragen mit Mehrfachauswahl Feedback aller Teilnehmer (auch anonymisiert) erhalten und direkt auswerten können. Über diese elektronisch unterstützten Feedbacksysteme können einerseits Aufmerksamkeit und Motivation der Lernenden positiv beeinflusst werden, andererseits haben Lehrende die Gelegenheit, ihren Unterricht auf die so erhaltene Resonanz abzustimmen.

Überdies können administrative Tätigkeiten vereinfacht werden und mobile Technologien beispielsweise für Anwesenheitsdokumentation, Überprüfen der Benotung, eine effizientere Zeitplanung oder zur Kommunikation mit den Eltern eingesetzt werden [20].

In Tabelle 2.2 sind die verschiedenen Sichtweisen in einer Zusammenfassung dargestellt.

2.3 Potenziale

Abschnitt 2.2 lässt bereits Rückschlüsse auf mögliche Potenziale von Mobile Learning zu, welches als wichtige Ergänzung zu konventionellen Lernmethoden gesehen werden kann. Im Folgenden wird auf diese und weitere Vorteile, welche sich durch mobiles Lernen ergeben, eingegangen:

2.3.1 Technologischer Fortschritt

Erst durch technologische Entwicklungen, wie sie in den letzten Jahren stattgefunden haben, wurde mobiles Lernen überhaupt möglich. Vor allem Smartphones gelten als perfekte Plattform für diese Art der Weiterbildung, weil

<i>Sichtweise</i>	<i>Zusammenfassung</i>
Individuell	Vorrangiges Ziel ist es, die Lernenden beim Wissensaufbau zu unterstützen, eine angemessene Lernumgebung anzubieten und eigenständiges Explorieren zu ermöglichen.
Situiert	Der Lernvorgang wird in Verbindung mit dem Kontext, in welchen er eingebettet ist, betrachtet. Das Umfeld spielt eine wichtige Rolle und Materialien sollten dementsprechend abgestimmt werden.
Kollaborativ	In Gruppen kann voneinander gelernt werden, wobei ein Austausch über verschiedene Sichtweisen stattfindet und eine dauerhafte Speicherung gefördert wird.
Informell und lebenslang	Lernen findet häufig in alltäglichen Situationen statt und die heutige Wissensgesellschaft macht es notwendig, Wissen und Fähigkeiten laufend zu aktualisieren.
Unterstützend	Administrations- und Kommunikationsprozesse von und zwischen Lehrenden und Lernenden können durch mobiles Lernen unterstützt und effizienter gestaltet werden.

Tabelle 2.2: Zusammenfassung der Sichtweisen auf Lernen.

eine große Anzahl von Sensoren sowie Verbindungstechnologien verfügbar sind. Letztere waren zentrales Thema eines aktuellen Berichts [33] des Forschungsinstitutes Berg Insight, auf welchem folgende Zahlen basieren:

Bluetooth⁴ ist auf praktisch allen Smartphones verfügbar, während jeweils 95 Prozent W-LAN⁵ sowie GPS⁶ Funktionalität besitzen. Mittlerweile beginnt sich auch die NFC⁷-Technologie allgemein durchzusetzen, was sich vor allem dadurch bemerkbar macht, dass die Verkäufe weltweit von geschätzten drei Millionen NFC-fähigen Smartphones 2010 auf rund 30 Millionen Einheiten im Jahr 2011 angestiegen sind.

Auch softwareseitig lassen aktuelle Smartphones wenig zu wünschen übrig. Medien wie Audio, Video oder Bilder lassen sich problemlos anzeigen beziehungsweise abspielen. Sollte dies nicht der Fall sein, können Anwender

⁴Bluetooth ermöglicht eine Funkverbindung über kurze Distanzen, vor allem zwischen wenigen Endgeräten.

⁵W-LAN (Wireless Local Area Network) bezeichnet lokale Funknetze, die meist im Infrastrukturmodes genutzt werden. Dabei übernimmt ein besonders ausgezeichnetes Gerät die Koordination anderer Netzknoten und stellt beispielsweise eine Internetverbindung zur Verfügung.

⁶GPS (Global Positioning System) wird bei der Positionsbestimmung eingesetzt und ist Voraussetzung für die meisten standortbezogenen Dienste.

⁷NFC (Near Field Communication) dient dem Austausch von Informationen über sehr kurze Strecken (nicht mehr als wenige Zentimeter) und wird beispielsweise bei bargeldlosen Zahlungen eingesetzt.

zusätzliche Software für diese oder andere Zwecke ohne großen Aufwand aus meist herstellerspezifischen Märkten wie dem Google Play Store oder dem App Store von Apple beziehen. Diese bieten gleichzeitig die Möglichkeit, selbst entwickelte Applikationen für andere zur Verfügung zu stellen, was für Anbieter von Mobile Learning Applikationen durchaus relevant ist.

2.3.2 Hoher Verbreitungsgrad

Auch die Verfügbarkeit von Mobiltelefonen in der Bevölkerung stellt ein wichtiges Potenzial für mobiles Lernen dar. Wie Keegan [12, S. 1] formuliert, sind vor allem die Technologien im Fernunterricht erfolgreich, welche bereits einen fixen Bestandteil im Leben der Anwender darstellen:

The history of distance education teaches us that it is technologies that are generally available to citizens which are successful in distance education, not those that have inherent didactic advantages.

Laut Weger [37] gab es in Österreich bereits Mitte 2006 mehr Handys als Einwohner. Weltweit geht der Trend in die gleiche Richtung, wie weitere Zahlen von Berg Insight [33] zeigen. Ende 2011 waren bereits 4,7 Milliarden Mobiltelefone (knapp 60 Prozent der Weltbevölkerung besitzen bereits mindestens ein solches) in Benutzung, wovon ein bedeutender Teil, 750 Millionen, aus Smartphones bestand. Deren Marktanteil ist in Ländern wie Österreich noch stärker ausgeprägt und befindet sich nach wie vor im Wachstum. Weltweit blieben 2011 die Verkäufe von normalen Mobiltelefonen ungefähr gleich, bei Smartphones erhöhten sich die Verkäufe hingegen um 60 Prozent.

2.3.3 Zeit- und Ortsunabhängigkeit

Die inhärente Eigenschaft von mobilen Geräten, problemlos mitgenommen werden zu können, befähigt deren Benutzer, diese jederzeit und an jedem Ort zur Verfügung zu haben. Durch die Unabhängigkeit und Flexibilität mobilen Lernens ist ein Zugriff auf zusätzliche Informationen genau in jenen Situationen möglich, in denen unmittelbar Bedarf besteht.

Darüber hinaus entstehen im Alltag immer wieder Pausen und Leerzeiten, wie beispielsweise beim Warten auf den Bus oder während längerer Zugfahrten. Mobile Learning erlaubt es, diese Gelegenheiten sinnvoll zu nutzen und sich dabei zusätzliches Wissen anzueignen. Dabei haben Oppermann und Specht [21, S. 2] jedoch folgende Bedenken:

Lernen wird in solchen Phasen zufälliger Leerzeiten weniger konzentriert, weniger zusammenhängend und das eingesetzte mobile Endgerät auf mittlere Sicht weniger komfortabel zu nutzen sein. Im Sinne eines geplanten zielorientierten [sic!] Lernens kann

es nicht das zukunftsweisende Konzept an M-Learning sein, zwischen Arbeitsende und Einkaufen noch schnell »eine Scheibe Lernen« an der Bushaltestelle einzuschieben.

In einem Großteil der Literatur wird das Nutzen der Leerzeiten aber dennoch als Vorteil hervorgehoben, wobei jedoch auch auf Einschränkungen, welche in dieser Arbeit in Abschnitt 2.4 zu finden sind, eingegangen wird. Krauss-Hoffmann und Kuszpa [14] führen in diesem Zusammenhang die mögliche Unterstützung repetitiven Lernens an, wobei der Lernende kurze Wiederholungsphasen zur Festigung der Lerninhalte nützen kann, beispielsweise beim Lernen von Sprachen.

2.3.4 Lernen im authentischen Kontext

Wie bereits in Abschnitt 2.2.2 näher erläutert, ist das Berücksichtigen des Kontextes, in welchem der Lernvorgang eingebettet ist, von hoher Wichtigkeit. Moderne Sensorik erlaubt es, eine Vielzahl von Informationen über die Umgebung und die Situation, in der sich die Lernenden befinden, zu generieren und dadurch Materialien kontextualisiert zur Verfügung zu stellen. Diese Integration des Lernens in seine zeitlichen, örtlichen und sozialen Rahmenbedingungen fördert eine individuellere Lernerfahrung und zählt zu den größten Stärken mobilen Lernens, worauf auch folgende Formulierung von Benford [3, S. 2] schließen lässt:

Users with mobile displays move through the world. Sensors capture information about their current context, including their location, and this is used to deliver them an experience that changes according to where they are, what they are doing, and maybe even how they are feeling. As a result, the user becomes unchained from their PC and experiences digital media that is interwoven with the everyday world, and that is potentially available in any place and at any time.

Bei real existierenden Lerngegenständen (wie beispielsweise Pflanzen, Maschinen oder Gebäuden) kann durch Mobile Learning ein Bezug zur jeweiligen Umgebung hergestellt werden. Dies erlaubt eine Erkundung der Lerngegenstände im authentischen Kontext, wobei Werkzeuge mobilen Lernens Instruktion, Recherche und Dokumentation ermöglichen [21].

2.4 Einschränkungen und Eigenheiten

Neben den bereits erwähnten Potenzialen birgt Mobile Learning Gefahren in sich, die bei der Gestaltung von Anwendungen und Lernmaterialien berücksichtigt werden müssen. Dies führt dazu, dass bereits existierende Software

und Inhalte unter Berücksichtigung der spezifischen Einschränkungen neu evaluiert und aufbereitet werden sollten, um die Lernenden bei ihren Aufgaben didaktisch und technisch weitestgehend zu unterstützen.

2.4.1 Technologische Barrieren

Trotz der raschen Weiterentwicklung unterliegen mobile Endgeräte den im E-Learning-Bereich etablierten Desktop-PCs in einigen Bereichen. Den wohl offensichtlichsten Nachteil stellen kleinere Gerätegrößen dar, welche jedoch für die an mobiles Lernen gestellten Mobilitätsanforderungen unumgänglich sind. Im Hinblick auf die kleineren Displays sind insbesondere Ressourcen mit längeren Textinhalten weniger geeignet [27], weshalb diese Unterlagen beispielsweise mit rich media objects⁸ angereichert werden sollten, um eine größere Menge von Informationen kompakt darstellen zu können. Ebenfalls wirkt sich die Größe einschränkend auf die verfügbaren Tasten und Schalter aus, welche bei den Geräten in verschiedener Anzahl und Anordnung vorhanden sind oder komplett durch Touchscreens in Verbindung mit softwareseitigen Lösungen ersetzt werden. Dies stellt sich besonders bei ungeübten Anwendern als problematisch dar, weshalb die Menge einzugebender Informationen, insbesondere längerer Texte, gering gehalten werden sollte.

Als weitere Barriere können Probleme mit der Energiezufuhr gewertet werden, die vor allem auf beschränkte Ladekapazität und steigenden Energieverbrauch der Endgeräte zurückzuführen sind. Für den Mobilitätsfaktor ist eine vom Stromnetz unabhängige Verwendungsmöglichkeit aber unumgänglich, weshalb beim Erwerb von Geräten auf eine möglichst hohe Ladekapazität geachtet werden sollte. Akkumulator-Technologien werden zwar laufend verbessert und vorhandene Kapazitäten steigen, gleichzeitig erhöhen zusätzliche und erweiterte Technologien jedoch laufend den Energieverbrauch, sodass die Akkulaufzeit im Endeffekt weitgehend unverändert bleibt.

2.4.2 Geminderte Konzentrationsfähigkeit

Die Ortsunabhängigkeit von Mobile Learning bringt Nebeneffekte mit sich, welche nicht nur positive Auswirkungen, wie beispielsweise beim Lernen im authentischen Kontext, haben. Das Umfeld, in welchem die Lernerfahrung stattfindet, kann sich auf die Konzentrationsfähigkeit durch Ablenkungen auch sehr negativ auswirken. Dies berichtet beispielsweise Kuszpa [15, S. 3], basierend auf Erfahrungen mit Studenten:

Beim Lernen an jedem Ort der Welt und zu jeder Zeit muss jedoch die Lernumgebung berücksichtigt werden. Die Studenten können sicherlich ruhiger in einem Park oder im Hotelzimmer

⁸Rich media objects sind mediale Inhalte wie beispielsweise Grafiken, Videos, Audios oder Animationen.

lernen als zum Beispiel im Bus oder Zug, wo unvermeidlich viele Geräusche und optische Ablenkungen herrschen.

Wird davon ausgegangen, dass die Lernenden durch das Umfeld zusätzlich belastet werden, sollten die Materialien wenn möglich so aufbereitet werden, dass sie auch mit geminderter Konzentrationsfähigkeit bearbeitet werden können. In solchen Situationen ist die Nachbearbeitung von bekannten Inhalten oder ein Wiederholen des Lernmaterials sinnvoll, weil umgebungsbezogene Einschränkungen das Erlernen von neuen, komplexen Inhalten unnötig erschweren würden. Diese Art von Aufgaben sollte an ruhigen, lernförderlichen Orten durchgeführt werden, oder generell dort, wo sich Potenziale, wie Lernen im authentischen Kontext, entfalten können.

2.4.3 Aufbereitung der Lernmaterialien

Zahlreiche Organisationen haben in den letzten Jahren auf E-Learning gesetzt und eigene Lösungen wie auch Inhalte produziert. Um nun eine hohe Akzeptanz von mobilem Lernen zu fördern und die Einstiegsschwierigkeiten niedrig zu halten, wäre es vorteilhaft, bestehende Inhalte und Strukturen ohne großen Aufwand übernehmen zu können. Dies erweist sich aber laut Krauss-Hoffmann und Kuszpa [14, S. 41] als schwierig:

Aufgrund der völlig anderen (technischen) Beschaffenheit von mobilen Geräten und ihren teilweise abweichenden, aber auch zusätzlichen Funktionalitäten, müssen notwendigerweise mittelfristig bekannte Ansätze zur Gestaltung von Lernangeboten für Mobile Learning kritisch beleuchtet und gegebenenfalls völlig neue Ansätze [...] konzipiert werden.

Aus diesem Grund ist eine erneute Evaluierung von bestehenden Architekturen und Materialien leider unumgänglich, um eine optimale Grundlage für didaktisch und technologisch sinnvolles mobiles Lernen schaffen zu können.

2.5 Bedeutsamkeit und Definition der Kontextinformationen

Wie bereits in Abschnitt 2.1.1 erwähnt, schöpft einfaches, sequentielles Übermitteln von Unterlagen an den Benutzer die Möglichkeiten mobilen Lernens bei weitem nicht aus. Aus diesem Grund ist es notwendig, zusätzliche Informationen, welche über den Lernvorgang und ihn umgebende Faktoren zur Verfügung stehen, zu verwerfen. Diese erlauben es, Lernmaterialien auf die Lernenden und deren Vorwissen abzustimmen, den Übertragungszeitpunkt

flexibel auszuwählen und somit situiertes Lernen unter möglichst optimalen Voraussetzungen zu ermöglichen. Kontextsensitive Anwendungen bieten somit einen benutzerspezifischen Mehrwert im Vergleich zu klassischen Applikationen.

An dieser Stelle bietet sich eine genauere Definition des Begriffs Kontext an, welche von Day [7, S. 3] geschaffen und in einer Vielzahl von Publikationen aufgegriffen wurde:

Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.

Unter Kontext werden dementsprechend jegliche Informationen über Entitäten verstanden, die in Zusammenhang mit der Interaktion zwischen Benutzer und Anwendung stehen. Diese Formulierung ist bewusst sehr allgemein gewählt, weshalb für den praktischen Einsatz detailliertere Darstellungen notwendig werden. Eine etwas ausführlichere Definition stammt von Zimmermann et al. [31] und wird im folgenden Abschnitt näher dargestellt.

2.6 Relevante Kategorien von Kontextinformationen

In ihrer Arbeit über eine operationale Definition von Kontext haben Zimmermann et al. [31] Informationen, welche den Kontext beschreiben, in fünf Kategorien eingeteilt: *Individualität*, *Zeit*, *Standort*, *Aktivität* und *Beziehungen*. Diese sind in Abbildung 2.2 dargestellt und werden anschließend detaillierter erläutert. Wie aus der Grafik ersichtlich ist, sind die Grenzen zwischen den einzelnen Gruppen verschwommen, sie beziehen sich aber alle auf die Entität, welche sie näher definieren.

Individualität

Beim Individualitäts-Kontext handelt es sich um Daten, welche die Entität selbst näher beschreiben. In Abhängigkeit vom Typ der beschriebenen Entität variieren diese Informationen, wobei Zimmermann et al. [31] grundsätzlich zwischen vier Arten von Entitäten unterscheiden, die in Tabelle 2.3 näher beschrieben werden.

Der Individualitäts-Kontext enthält somit all jene Informationen, die sich bei einer Entität oder einer Gruppe von Entitäten beobachten lassen und deren Zustand näher beschreiben.

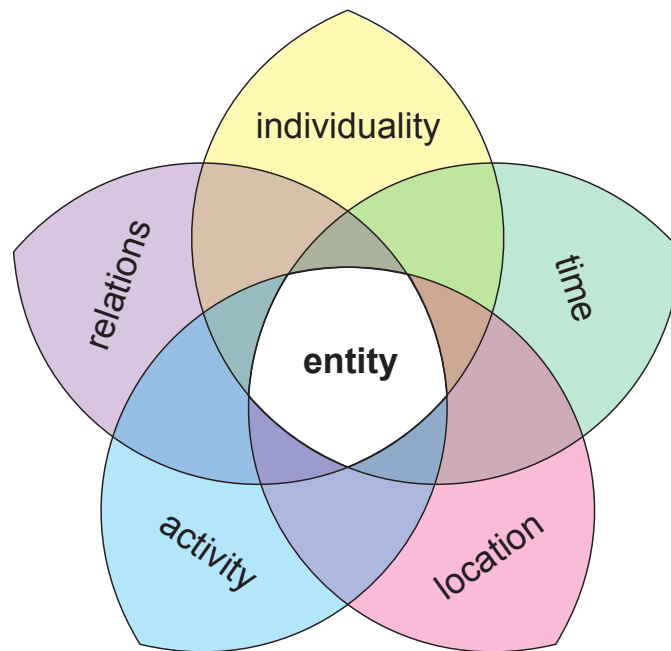


Abbildung 2.2: Fünf fundamentale Kategorien von Kontextinformationen [31].

Zeit

Zeitliche Informationen spielen in kontextsensitiven Anwendungen eine wichtige Rolle. Diese Kategorie ermöglicht es, Daten mit einem Zeitstempel oder einer Gültigkeitsdauer zu versehen, was eine Vielzahl an Möglichkeiten eröffnet. Durch eine Verknüpfung mit Standortdaten kann beispielsweise die bisherige Route des Anwenders nachvollzogen werden. Dieses Beispiel zeigt darüber hinaus, wie eng die Gruppen von Kontext-Kategorien in Verbindung miteinander stehen – wie in diesem Fall der Zeit- und Standort-Kontext.

Ebenfalls zu dieser Kategorie gehören Daten über die Zeitzone des Anwenders oder die aktuelle Systemzeit. Insbesondere im Hinblick auf das Erstellen und Verwerten einer Historie von Datenbeständen müssen gespeicherte Informationen zeitlich markiert werden.

Standort

Moderne mobile Geräte, insbesondere Smartphones, besitzen im Normalfall bereits Technologien, welche deren aktuelle Position feststellen lassen (siehe 2.3.1). Diese Daten erlauben es standortbezogenen Anwendungen, Entitäten in der näheren Umgebung herauszufiltern und dem Anwender zu präsentieren. Des Weiteren fallen Richtungs- und Geschwindigkeitsinformationen in diese Kategorie, welche Aufschluss über den zukünftigen Standort oder

<i>Entität</i>	<i>Beschreibung</i>
Natürlich	Dieser Kategorie gehören die Charakteristika aller Dinge an, die keiner menschlichen Handlung entstammen und somit natürlichen Ursprungs sind. Als Beispiele können Informationen über Tiere, Mineralien oder Sehenswürdigkeiten natürlichen Ursprungs, wie etwa den Großglockner, genannt werden.
Menschlich	In diese Gruppierung fallen Daten über Menschen, wie Nutzerverhalten, bevorzugte Sprache oder deren emotionaler Zustand.
Künstlich	Diese Informationen beziehen sich auf Entitäten, die aus menschlichen Handlungen resultieren. Gebäude, Bücher, Hardware und Lernunterlagen sind nur ein minimaler Ausschnitt der möglichen Entitäten.
Gruppen	Daten über Entitäten, die ähnliche Eigenschaften teilen und miteinander agieren, können zu Gruppen zusammengefasst werden. Entitäten können in keiner, einer oder mehreren Gruppen existieren und als mögliche Informationen können Interessen, Fähigkeiten oder bei technischen Geräten auch die Auflösung genannt werden.

Tabelle 2.3: Arten von Entitäten.

Hinweise auf die aktuelle Tätigkeit des Benutzers, wie beispielsweise Warten oder Fahren, geben können.

Informationen über den Standort können dafür verschieden kodiert werden, wobei ein häufig verwendeter Ansatz durch die Verwaltung von Längen- und Breitengrad sowie Höhe umgesetzt wird. Für Positionen müssen nicht zwingend absolute Werte angenommen werden, es ist auch möglich, den Standort relativ zu anderen Entitäten anzugeben.

Aktivität

Um diese Gruppe von Kontextinformationen näher beschreiben zu können, ist eine genauere Definition des Begriffs Aktivität nötig. Zimmermann et al. [31, S. 6] umschreiben diesen wie folgt:

An entity's activity determines to a great extend its current needs. The activity context covers the activities the entity is currently and in future involved in and answers the question „What does the entity want to achieve and how?“

Der Aktivitäts-Kontext beschäftigt sich somit vor allem mit den Zielen, welche die jeweilige Entität gerade hat. Hier sind insbesondere die Absichten

des Anwenders von Interesse, welche mithilfe der kontextsensitiven Applikation gelöst werden sollen. Zimmermann et al. [31] unterscheiden zwischen Zielen niedriger und höherer Ebene, wobei erstere häufig variieren und letztere vergleichsweise konsistent bleiben.

Beziehungen

Entitäten stehen nicht für sich allein, sondern können Beziehungen zu anderen Entitäten aufbauen. Zwischen zwei Entitäten können mehrere verschiedene Verbindungen bestehen, welche sich im Laufe ihrer Existenz weiterentwickeln oder wieder verschwinden können. Diese Beziehungen können eine Vielzahl verschiedener Ausprägungen annehmen, welche von Zimmermann et al. [31] in soziale, funktionale und kompositionale Beziehungen unterteilt werden:

Soziale Beziehungen: Diese Art von Beziehungen beschreibt in der Regel interpersonale Verbindungen zwischen zwei oder mehreren Personen, wie Verbindungen zu Freunden, Verwandten oder auch Feinden.

Funktionale Beziehungen: Entitäten können von anderen zu einem bestimmten Zweck in Anspruch genommen werden, wie beispielsweise beim Lesen einer Seite durch den Anwender, um gewisse Informationen zu erhalten.

Kompositionale Beziehungen: Entitäten können aus anderen aufgebaut sein. So besteht beispielsweise ein Kapitel aus mehreren Seiten und kann wiederum mit anderen Kapiteln zu einem Buch kombiniert werden.

Kapitel 3

Informationsstrukturierung und -auswahl

Um Inhalte für Mobile Learning nützen zu können, sollten diese speziell aufbereitet werden, damit sie bei der späteren Auswahl individuell auf die Lernenden und deren Umfeld abgestimmt werden können. Dazu müssen neben den Lernobjekten auch Daten über das Umfeld und den Lernenden, wie Vorwissen, Interessen oder dessen Position in gut verarbeitbarer Form verwaltet werden. In dieser Arbeit wird für die Administration solcher Daten ein ontologiebasierter Ansatz verfolgt, der eine effektive Grundlage für individualisierte Informationsauswahl darstellt. In diesem Kapitel werden Technologien vorgestellt, welche ontologiebasierte Strukturierung und Informationsauswahl ermöglichen und als Basis für den in Kapitel 5 und 6 vorgestellten Lösungsansatz dienen. In Abschnitt 3.2 werden darüber hinaus alternative Ansätze vorgestellt.

3.1 Grundlagen Ontologien

Eine Datenbasis, welche aus den in Abschnitt 2.5 beschriebenen Kontextinformationen aufgebaut ist, kann als Grundlage für Mobile Learning Applikationen verwendet werden. Alle benötigten Inhalte, wie Informationen über die Lernenden, deren Interessen und Vorwissen, für sie unbekannte oder bereits abgerufene Lernobjekte und alle weiteren essentielle Daten, können durch Entitäten und die verschiedenen Kategorien von Kontextinformationen dargestellt werden. Der in dieser Arbeit verfolgte ontologiebasierte Ansatz lässt dieses Wissen in digitaler Form repräsentieren und bietet die Grundlage der weiteren praktischen Umsetzung.

3.1.1 Definition

Grundsätzlich dienen Ontologien im Verständnis der Informatik einer formalen Beschreibung der Struktur eines Systems. Insbesondere das Forschungsgebiet der künstlichen Intelligenz hat zu deren Entwicklung beigetragen, weil es für viele Fragestellungen in diesem Bereich unumgänglich ist, Wissen in maschinenlesbarer Form zu repräsentieren. Studer et al. [29, S. 25] haben des Weiteren eine häufig zitierte Definition dieses Begriffs geschaffen:

An ontology is a formal, explicit specification of a shared conceptualisation.

Die Autoren definieren in dieser Publikation die von ihnen verwendeten Begriffe näher: *Formal* weist darauf hin, dass die erstellte Ontologie maschinenlesbar sein muss. *Explicit* drückt aus, dass die Art der verwendeten Konzepte sowie mögliche Einschränkungen auf deren Verwendung genau definiert sein müssen. *Conceptualisation* beschreiben die Autoren so, dass von einem in der echten Welt vorkommenden Phänomen ein abstraktes Modell erstellt wird, welches die relevanten Konzepte beinhaltet. *Shared* bedeutet, dass in einer Ontologie verwaltetes Wissen von einer Gruppe akzeptiert wird und nicht nur privates Wissen eines Individuums abgebildet ist [29].

Ontologien ermöglichen die Verwendung von Konzepten, welche für die Verwaltung von Kontextinformationen essentiell sind. Sie umfassen Klassen, deren Attribute, sowie Beziehungen zwischen ihnen. Klassen erlauben beispielsweise die Repräsentation des Lernenden, während Attribute diesen näher definieren und Beziehungen die Verbindungen zu anderen Klassen abbilden. Als Beispiel für ein Attribut wäre der Name des Lernenden geeignet und „interessiert_an“ könnte als Beziehung zu einer Klasse definiert sein, die zur Darstellung von Interessen der Lernenden eingesetzt wird. Darüber hinaus ist es möglich, aus vorhandenem Wissen durch Regeln logische Schlussfolgerungen zu ziehen [1].

3.1.2 Web Ontology Language

Um Ontologien formal abbilden, publizieren und verteilen zu können, werden spezielle Beschreibungssprachen verwendet. Die Web Ontology Language (OWL) wurde im Februar 2004 vom W3C¹ als Ontologiesprache standardisiert, wobei während der Entwicklung auf ein gutes Gleichgewicht zwischen Ausdrucksstärke der Sprache und effizientem Schlussfolgern geachtet wurde

¹Das World Wide Web Consortium (W3C) beschäftigt sich mit der Entwicklung von Standards für das World Wide Web und Technologien, welche dieses betreffen.

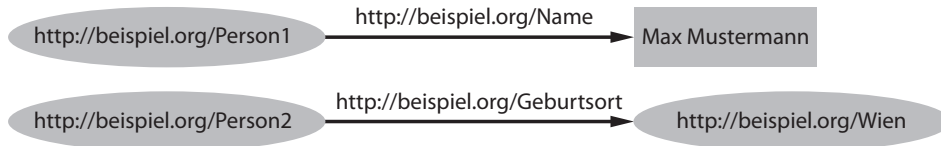


Abbildung 3.1: Grafische Darstellung eines Tripel.

[10]. OWL ist von DAML+OIL² abgeleitet und baut auf RDF³ und RDFS⁴ auf, wobei erweiterte Ausdrucksmittel zur Verfügung stehen und komplexere Zusammenhänge ausgedrückt werden können.

Im Folgenden werden für die nächsten Kapitel wichtige Konzepte näher erläutert. OWL unterstützt im Gegensatz zu RDF und RDFS alle vorgestellten Elemente.

Tripel: Auf unterster Ebene werden Tripel eingesetzt, um die zu repräsentierenden Informationen abbilden zu können. Ein Tripel beschreibt jeweils Subjekt, Prädikat und Objekt, welche genau einer Kante des zu repräsentierenden RDF-Graphen entsprechen, der alle bekannten Informationen abbildet. Dieser lässt sich vollständig durch die Angabe seiner Kanten beschreiben [10].

In Abbildung 3.1 sind beispielhaft zwei Tripel abgebildet, wobei im ersten Beispiel `http://beispiel.org/Person1` als Subjekt verwendet wird, `http://beispiel.org/Name` das Prädikat ist und ein Literal⁵ mit dem Wert „Max Mustermann“ das Objekt darstellt. Um Subjekt, Prädikat und Objekt eindeutig identifizieren zu können, werden, wie in der Grafik erkennbar, URIs⁶ verwendet. Das erste dargestellte Tripel enthält als Information, dass *Person1* den Namen „Max Mustermann“ trägt, während im zweiten Tripel angegeben wird, dass der Geburtsort von *Person2* Wien ist.

²Diese Beschreibungssprache war eine Kombination aus DARPA Agent Markup Language (DAML) und Ontology Inference Layer (OIL), welche zugunsten von OWL nicht mehr vom W3C weiterentwickelt wird.

³Das Resource Description Framework (RDF) ist ebenfalls ein Standard des W3C und erlaubt die Beschreibung von Daten mithilfe eines gerichteten Graphen. Details zur Spezifikation können unter <http://www.w3.org/TR/rdf-concepts/> abgerufen werden.

⁴Auch das Resource Description Framework Schema wurde vom W3C entwickelt und erweitert RDF insofern, dass ein eigenes Vokabular für den Aufbau einfacher Ontologien eingeführt werden kann. Die Spezifikation ist unter <http://www.w3.org/TR/rdf-schema/> verfügbar.

⁵Literale werden zur Darstellung von Datenwerten, beispielsweise Zeichenketten oder numerische Werte, verwendet und werden in grafischen Darstellungen als Rechteck abgebildet.

⁶Uniform Resource Identifiers (URIs) werden zur eindeutigen Identifizierung von Ressourcen (Entitäten der realen Welt), vor allem im World Wide Web, verwendet und müssen einem festgelegten Aufbau folgen.

Ontologien können anhand dieser Tripel vollständig abgebildet werden, wodurch sie in speziellen Datenbanken, sogenannten Triplestores, abgespeichert und in weiterer Folge ausgewertet werden können.

Klassen, Eigenschaften und Individuen: Diese Konzepte dienen als Grundbausteine von OWL. Klassen stehen jeweils für eine Menge von Ressourcen, während Individuen die Instanzen dieser Klassen darstellen. Eigenschaften, auch Rollen genannt, gruppiert man zusätzlich in abstrakte und konkrete Eigenschaften, wobei erstere Individuen mit Individuen verbinden und letztere Verbindungen zwischen Individuen und Literalen darstellen [10].

Person1 und *Person2* aus Abbildung 3.1 könnten beispielsweise beide Individuen der Klasse *Person* sein. Hinsichtlich der Eigenschaften wäre der Name eine konkrete Eigenschaft, weil er direkt mit einem Literal verbunden ist. Der Geburtsort stellt hingegen eine abstrakte Eigenschaft dar und enthält deshalb einen Verweis auf ein anderes Individuum.

Vordefinierte Sprachelemente: Zur Strukturierung von Daten werden vordefinierte Sprachelemente eingesetzt, wodurch effizientes Schlussfolgern ermöglicht wird, was eine der besonderen Stärken von OWL ist. Sie werden im Folgenden angelehnt an eine Kategorisierung von Hitzler et al. [10] vorgestellt, wobei aus Gründen der Übersichtlichkeit nicht die komplette Liste der Sprachelemente dargestellt ist. Eine detailliertere Auflistung und Beschreibung kann jedoch in der soeben erwähnten Publikation oder in den Standards des W3C⁷⁸ nachgeschlagen werden. Tabelle 3.1 zeigt die Sprachelemente, welche Beziehungen zwischen Klassen und Individuen definieren können.

<i>Bezeichnung</i>	<i>Erläuterung</i>
A rdfs:subClassOf B	Alle Individuen der Klasse A sind gleichzeitig Individuen der Klasse B.
A owl:disjointWith B	Die Klassen A und B haben keine gemeinsamen Individuen.
A owl:equivalentClass B	Die Klassen A und B sind gleich.
C owl:sameAs D	Die Individuen C und D sind gleich.
C owl:differentFrom D	Die Individuen C und D sind verschieden.

Tabelle 3.1: Beziehungen zwischen Klassen und Individuen.

Logische Konstruktoren ermöglichen darüber hinaus weitere Schlussfolgerungsmechanismen und erlauben es, komplexeres Wissen auszudrücken. Sie ermöglichen das Abbilden von Konjunktion (logisches und), Disjunktion

⁷<http://www.w3.org/TR/rdf-schema/>

⁸<http://www.w3.org/TR/owl-ref/>

(logisches oder) und Negation (logisches nicht) und werden in Tabelle 3.2 näher erläutert.

<i>Bezeichnung</i>	<i>Erläuterung</i>
A owl:intersectionOf B	Umfasst alle Objekte, die gleichzeitig zu Klasse A und B gehören.
A owl:unionOf B	Umfasst alle Objekte der Klassen A und B.
A owl:complementOf B	Umfasst alle Objekte der Klasse A, ausgenommen der Objekte von Klasse B.

Tabelle 3.2: Logische Konstruktoren auf Klassen.

Durch Einschränkungen auf Eigenschaften werden weitere Konstrukturen zur Definition komplexer Klassen zur Verfügung gestellt. Diese sind in Tabelle 3.3 dargestellt.

<i>Bezeichnung</i>	<i>Erläuterung</i>
E owl:allValuesFrom X	Für Eigenschaft E sind nur Individuen der Klasse oder des Wertebereichs X als Wert zugelassen.
E owl:someValuesFrom X	Für Eigenschaft E muss mindestens ein Individuum der Klasse oder dem Wertebereich X entsprechen.
E owl:hasValue Y	Für Eigenschaft E wird bereits der konkrete Wert Y (Individuum oder Datenwert) angegeben.
E owl:minCardinality Z	Für Eigenschaft E muss die Mindestanzahl Z an verschiedenen Werten gegeben sein.
E owl:maxCardinality Z	Für Eigenschaft E ist die Anzahl verschiedener Werte auf das Maximum Z beschränkt.
E owl:cardinality Z	Für Eigenschaft E muss die genaue Anzahl Z verschiedener Werte gegeben sein.

Tabelle 3.3: Einschränkungen auf Eigenschaften.

Es ist ebenfalls möglich, Eigenschaften aufeinander zu beziehen. Sprach-elemente, welche dies ermöglichen, sind in Tabelle 3.4 angeführt. Zusätzlich werden in dieser Tabelle zwei Merkmale der Eigenschaften, rdfs:range und rdfs:domain, vorgestellt.

<i>Bezeichnung</i>	<i>Erläuterung</i>
E rdfs:subPropertyOf F	Die Eigenschaft E ist die nähere Beschreibung der Eigenschaft F.
E owl:equivalentProperty F	Eigenschaft E ist gleich der Eigenschaft F.
E owl:inverseOf F	Eigenschaft E beschreibt dieselbe Relation wie Eigenschaft F, jedoch mit vertauschten Argumenten.
E rdfs:range X	Für Eigenschaft E ist nur die Klasse oder der Wertebereich X als Objekt zulässig.
E rdfs:domain A	Bindet die Eigenschaft E an eine Klasse A, welche das Subjekt darstellen muss.

Tabelle 3.4: Beziehungen und Merkmale von Eigenschaften.

3.1.3 SPARQL

Um Informationen aus einer ontologiebasierten Datenbasis extrahieren zu können, kann SPARQL⁹ eingesetzt werden. SPARQL ist eine standardisierte Abfragesprache für RDF-Graphen und wird vom W3C entwickelt, um gezielt Informationen aus Datenquellen wie Triplestores selektieren zu können. Grundsätzlich wird für SPARQL-Abfragen folgende Syntax angewendet (das Beispiel entstammt [10]):

```

1 PREFIX ex: <http://example.org/>
2 SELECT ?titel ?autor
3 WHERE
4   { ?buch ex:VerlegtBei <http://springer.com/Verlag> .
5     ?buch ex:Titel ?titel .
6     ?buch ex:Autor ?autor . }
```

In der ersten Zeile wird durch das Schlüsselwort *PREFIX* ein Namensraum definiert, wodurch das angegebene Kürzel in der nachfolgenden Abfrage anstelle des längeren IRI¹⁰ verwendet werden kann. Das Schlüsselwort *SELECT* gibt Aufschluss über das Ausgabeformat und kann an dieser Stelle neben *CONSTRUCT*, *ASK* sowie *DESCRIBE* eingesetzt werden. Die verschiedenen Ausgabeformate sind in Tabelle 3.5 näher beschrieben.

Der wichtigste Teil der Abfrage wird durch die *WHERE*-Klausel eingeleitet, wonach, eingeschlossen in geschweifte Klammern, ein einfaches Graph-Muster angegeben wird. Dieses besteht aus Tripeln, welche zusätzlich Variablenbezeichner (angegeben durch *[/Variablenname/]*) enthalten können. Solche Platzhalter werden erst durch die Abfrage mit konkreten Werten befüllt

⁹SPARQL ist ein rekursives Akronym und steht für SPARQL Protocol and RDF Query Language. Details zur Spezifikation sind unter [36] verfügbar.

¹⁰Internationalized Resource Identifier (IRI) stellen die internationalisierte Form des Unique Resource Identifier (URI) dar und dienen zur Identifizierung von Ressourcen, vor allem im Internet.

<i>Schlüsselwort</i>	<i>Ausgabeformat</i>
SELECT	Liefert eine Liste von Variablenbelegungen zurück, welche über Programmierschnittstellen gut weiterverarbeitet werden können. Hierbei wird entweder eine Liste von Variablennamen übergeben oder es wird durch das Symbol * angegeben, dass alle in der Abfrage vorkommenden Variablen zurückgeliefert werden sollen.
CONSTRUCT	Ermöglicht es, dass mehrere Tripel in Form eines Graphen als Ergebnis zurückgegeben werden. Dieser Graph kann beispielsweise selbst in einen Triplestore eingepflegt werden oder als Grundlage für weitere Abfragen dienen.
ASK	Überprüft, ob das angegebene Muster gefunden werden kann und gibt basierend auf diesem Ergebnis „wahr“ oder „falsch“ zurück.
DESCRIBE	Kann wie <i>SELECT</i> verwendet werden (mit Variablen und dem Symbol *), gibt jedoch Beschreibungen zu den übergebenen Ressourcen zurück. Dies ist beispielsweise dann sinnvoll, wenn eine RDF-Datenbank durchsucht werden soll, über deren interne Struktur nur wenige Informationen vorliegen.

Tabelle 3.5: Mögliche Ausgabeformate bei SPARQL-Abfragen [36].

und können mit demselben Bezeichner öfter angegeben werden, was bedeutet, dass an diesen Stellen das gleiche Element auftreten muss. Das Graph-Muster wird mit dem Datengraphen abgeglichen um passende Ergebnisse auszuwählen [10].

Im obigen Beispiel werden somit alle Dinge ausgewählt, die beim Springer Verlag verlegt wurden und über die Eigenschaften *ex:Titel* und *ex:Autor* einen Titel sowie Autor zugewiesen bekommen haben. Für jedes so gefundene „Ding“ wird ein Variablenpaar, bestehend aus dessen Titel und Autor, zurückgegeben. Um die Abfrage explizit auf Bücher einzuschränken, kann zusätzlich ein Tripel angegeben werden, das den Typ von *?buch* mit *rdf:type* festlegt:

```

1 PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX ex: <http://example.org/>
3 SELECT ?titel ?autor
4 WHERE
5   { ?buch ex:VerlegtBei <http://springer.com/Verlag> .
6     ?buch ex:Titel ?titel .
7     ?buch ex:Autor ?autor .
8     ?buch rdf:type ex:Buch . }
```

Mit dieser Basisfunktionalität können bereits viele Abfragen durchgeführt werden. Für komplexere Abfragemuster stehen jedoch weitere Ausdrucksmittel zur Verfügung, wobei für die praktische Umsetzung des in Kapitel 5 und 6 präsentierten Frameworks vor allem das Verständnis der Schlüsselwörter *OPTIONAL* sowie *FILTER* und ihrer Arbeitsweise notwendig ist. Diese werden anhand des nachfolgenden Beispiels (angelehnt an Beispiele aus [10]) erklärt:

```

1 PREFIX ex: <http://example.org/>
2 PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3 SELECT ?titel ?autor WHERE
4   { ?buch rdf:type ex:Buch .
5     ?buch ex:VerlegtBei <http://springer.com/Verlag> .
6     ?buch ex:Titel ?titel .
7     ?buch ex:Preis ?preis
8     FILTER (?preis < 35)
9     OPTIONAL { ?buch ex:Autor ?autor }
10  }
```

OPTIONAL: Dieses Schlüsselwort ermöglicht es, dass danach angegebene Graph-Muster nicht zwingend in allen ermittelten Ergebnissen vorkommen müssen. Sie können jedoch, wenn sie gefunden werden, das Ergebnis erweitern und zur Bindung zusätzlicher Variablen führen [10]. Im angegebenen Beispiel müssen für die ausgewählten Bücher somit keine Autoren angegeben sein, ist dies jedoch der Fall, ist auch die Variable *?autor* belegt.

FILTER: Um Ergebnisse zusätzlich einzuschränken, können Filter eingesetzt werden. Dem Schlüsselwort folgt eine Filterfunktion, die typischerweise Parameter besitzt und für jedes mögliche Ergebnis „wahr“ oder „falsch“ zurückliefert. Filter beziehen sich nicht strikt auf das RDF-Datenmodell, sondern können im Prinzip jede maschinell überprüfbare Bedingung enthalten [10]. Die im Beispiel angeführte Filterbedingung schränkt das Ergebnis auf solche Bücher ein, deren Preis weniger als 35 beträgt. Eine komplette Auflistung der verfügbaren Filtermöglichkeiten ist in der offiziellen SPARQL Dokumentation [36] angeführt.

Diese sehr kompakte Einführung zu SPARQL sollte ein grundsätzliches Verständnis der RDF-Abfragesprache und ihrer Syntax vermittelt haben. SPARQL wird in dieser Arbeit vor allem bei der serverseitigen Auswahl der Lernobjekte verwendet, die in Abschnitt 5.4 näher erläutert wird.

3.2 Alternativen

Ontologiebasierte Ansätze sind jedoch nicht die einzige Möglichkeit, Kontextinformationen zu verwalten. Besonders in den letzten Jahren wurden

verschiedene Methoden in Betracht gezogen und evaluiert, wobei sich große Unterschiede ergeben haben, wie exakt Konzepte der echten Welt repräsentiert werden können. In folgendem Abschnitt werden diese Konzepte zusammengefasst dargestellt und kurz diskutiert, wobei die grundsätzliche Kategorisierung auf einer Arbeit von Moore et al. [19] basiert.

Key-Value-Modelle

Diese überaus simple Methode stellt eher ein Konzept dar, wie Kontextinformationen grundsätzlich in einer Datenstruktur abgespeichert werden können. Dabei lassen sich mögliche Kontextinformationen als Attribute definieren, während deren konkrete Ausprägung als Wert zum jeweiligen Attribut abgespeichert wird. Folgendes Beispiel zeigt eine mögliche Wertzuweisung der Attribute Name und Position, welche hier als XML¹¹-Elemente angegeben werden:

```
<Name>Max Mustermann</Name>
<Position>48.36812500,14.51313520</Position>
```

Die Simplizität von Key-Value-Modellen führt bereits bei der Abbildung solcher Strukturen zu Problemen, bei denen Attribute Beziehungen zueinander aufweisen müssen. Dies ist zur Gestaltung der Kontextinformationen von Mobile Learning Applikationen jedoch unzureichend, weshalb dieser Ansatz als nicht geeignet angesehen werden kann.

Markup Scheme Modelle

Diese Art, Kontextinformationen zu strukturieren, wird durch die Verwendung von bestimmten Auszeichnungssprachen ermöglicht. Diese erlauben das Anreichern von Text mit zusätzlichen Informationen und können, beispielsweise bei XML oder HTML¹², zur hierarchischen Strukturierung von Daten genutzt werden, wie durch folgendes Beispiel gezeigt wird:

```
<Learner>
  <Name nickname="max">Max Mustermann</Name>
  <Position>
    <Latitude>48.36812500</Latitude>
    <Longitude>14.51313520</Longitude>
  </Position>
</Learner>
```

¹¹Die Extensible Markup Language (XML) ist eine Auszeichnungssprache und ermöglicht hierarchische Darstellungen von Daten.

¹²Die Hypertext Markup Language (HTML) dient als Auszeichnungssprache vorrangig der Strukturierung von Webseiten.

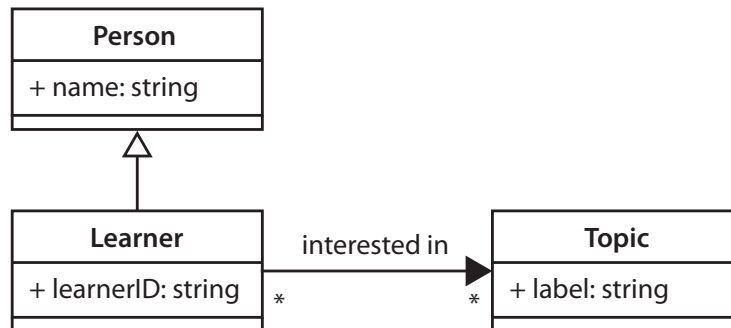


Abbildung 3.2: Beispielhafte objektorientierte Modellierung anhand eines Klassendiagramms.

Markup Scheme Models erlauben bereits fortgeschrittenere Strukturen als Key-Value-Modelle, sind jedoch für mobile Lernanwendungen nur sehr eingeschränkt verwendbar. Das ist darin begründet, dass sie für relativ spezifische Anwendungszwecke entwickelt wurden und im Hinblick auf Kontextinformationen nur begrenzt verwendet werden können [18].

Objektorientierte Modelle

Kontextinformationen auf diese Art zu modellieren erlaubt das Zurückgreifen auf Prinzipien der Objektorientierung, wie Kapselung und Vererbung, wodurch hohe Wiederverwendbarkeit der Objekte gewährleistet ist. Diese Methoden stellen laut Moore et al. [19] eine gute Möglichkeit dar, die hohe Dynamik von kontextsensitiven Anwendungen zu repräsentieren. In Abbildung 3.2 ist eine beispielhafte objektorientierte Modellierung anhand eines Klassendiagramms dargestellt.

Objektorientierte Modelle setzen jedoch voraus, dass Objekte in verschiedenen Anwendungen gleich aufgebaut sind, um Interoperabilität zu gewährleisten. Für offene und dynamische Umgebungen sind sie weniger geeignet [18].

Logikbasierte Modelle

In diesen Systemen werden Kontextinformationen durch Fakten, Ausdrücke und Regeln dargestellt. Neue oder veränderte Kontextinformationen werden dem System entweder als Fakten hinzugefügt oder mithilfe von Regeln logisch abgeleitet. Logikbasierte Modelle weisen einen hohen Grad an Formalismus auf [28].

Zur exemplarischen Darstellung eines logikbasierten Modells wird auf ein Beispiel von Strang und Linnhoff-Popien [28] zurückgegriffen, welches auf einem System von McCarthy [17] basiert. In diesem System kann eine

Beziehung beispielsweise durch

$$ist(c, p)$$

angegeben werden, was bedeutet, dass die Aussage p dann wahr ist, wenn der Kontext c gegeben ist. Dies erlaubt Angaben wie:

$$ist(context-of(„Sherlock Holmes stories“), „Holmes is a detective“)$$

Im Kontext von Geschichten über Sherlock Holmes gilt somit, dass Sherlock Holmes ein Detektiv ist.

Logikbasierte sowie objektorientierte Modelle bieten für sich alleine bereits eine gute Grundlage für kontextsensitive Systeme. Der in Abschnitt 3.1 vorgestellte Ansatz eines ontologiebasierten Modells vereint jedoch viele Stärken logikbasierter (zum Beispiel Regeln und damit verbundenes Ableiten von Fakten) und objektorientierter (beispielsweise Klassen, Individuals/Instanzen und Vererbung) Systeme, weshalb er als Grundlage für das in der weiteren Arbeit verfolgte und in Kapitel 5 und 6 vorgestellte System ausgewählt wurde.

Kapitel 4

Verwandte Arbeiten

In Zusammenhang mit mobilem Lernen wurden bereits Systeme entwickelt und in Publikationen behandelt, wobei insbesondere semantische Ansätze einer relativ neuen Generation angehören. In diesem Abschnitt werden verwandte Arbeiten mit einem ebenfalls ontologiebasierten Lösungsansatz vorgestellt.

4.1 m-LOCO: An Ontology-based Framework for Context-Aware Mobile Learning

Melody Siadaty et al. [26] entwickelten mit dem Ziel, personalisiertes und kontextsensitives Lernen in mobilen Lernumgebungen zu ermöglichen, ein ontologiebasiertes Framework. Dieses trägt den Namen *m-LOCO* und beruht auf dem ebenfalls ontologiebasierten Framework *LOCO*¹, welches bereits das Verwalten von Lernobjekten und mehreren Kontextinformationen erlaubt.

Der Kern von *LOCO* wird aus drei Ontologien gebildet. *Learning object content* beschreibt den Aufbau der Lernobjekte, welche die eigentliche Information beinhalten, während *learning design* pädagogische Informationen beinhaltet. Die dritte Kernontologie, *LOCO-Cite*, dient als Verlinkung zu anderen relevanten Ontologien wie *user modeling* (zur Verwaltung der Benutzerdaten) oder den *subject domain ontologies*, welche der formalen Beschreibung des Lernthemas dienen [26].

Die bestehenden Ontologien werden durch *m-LOCO* um Informationen, welche das Übertragungsmedium betreffen, erweitert, wie beispielsweise die Bildgröße oder Daten zur Netzwerkgeschwindigkeit, um etwaige Einschränkungen berücksichtigen zu können. Ebenfalls wird das System insofern ergänzt, dass es dem Benutzer verbesserte Möglichkeiten bietet, seinen Lernvorgang selbst zu regulieren und sich mit anderen Lernenden auszutauschen um gemeinsame Problemlösungen erarbeiten zu können [26].

¹Die Bezeichnung *LOCO* steht hierbei für Learning Object Context Ontologies.

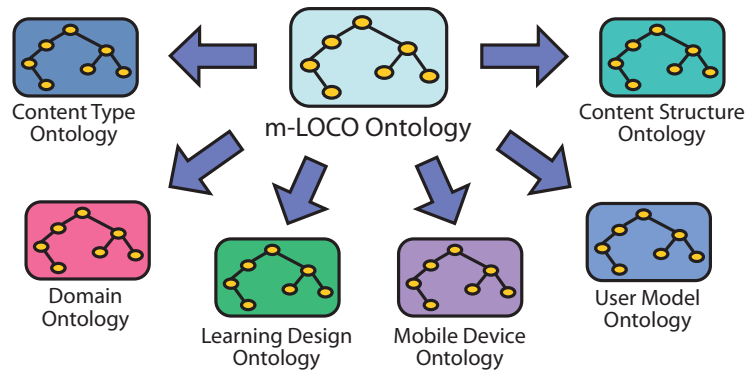


Abbildung 4.1: Die *m-LOCO* Ontologie inklusive Verbindungen [26].

In Abbildung 4.1 ist ein Überblick der von *m-LOCO* verwendeten Ontologien dargestellt. Die *Content Type* und *Content Structure Ontologien* dienen der Beschreibung des eigentlichen Lerninhalts und dessen Struktur, während die *Domain Ontology* das gesamte Themengebiet formal modelliert. Pädagogische Informationen werden durch die *Learning Design Ontology* verwaltet, Daten über das Übertragungsmedium finden hingegen in der *Mobile Device Ontology* Berücksichtigung. Der Lernende und dessen Kompetenzen, aber auch Lehrende und Autoren von Lerninhalten sind in der *User Model Ontology* abgebildet.

Bei *m-LOCO* ist vorgesehen, dass Lehrende die eigentlichen Inhalte aufbereiten und einpflegen, wohingegen die Auswahl dieser Informationen im laufenden Betrieb automatisch durch die Abfragesprache SPARQL erfolgt. Diese ist in Abschnitt 3.1.3 beschrieben und ermöglicht mithilfe der soeben vorgestellten Ontologien nicht nur die Abfrage von bestmöglich für den aktuellen Kontext geeigneten Lerninhalten, sondern beispielsweise auch das Finden von anderen Benutzern, die gerade ähnliche Aktivitäten durchführen und vergleichbare Kompetenzen aufweisen.

Das *m-LOCO*-Framework bietet sehr umfangreiche Möglichkeiten, Kontextinformationen und Inhalte, welche für Mobile Learning relevant sind, zu verwalten. Bereits dessen Vorgänger, das *LOCO*-Framework, findet praktische Verwendung in einem benachbarten Anwendungsfeld, was dessen praxisbezogene Relevanz aufzeigt (beispielsweise in *LOCO-Analyst*², einem Tool zur Evaluierung von E-Learning Maßnahmen). Im Hinblick auf den standortbezogenen Kontext, welcher (wie in Abschnitt 2.6 ausgeführt) von hoher Wichtigkeit für mobiles Lernen ist und auch in der in Kapitel 5 und 6 präsentierten Umsetzung einen besonderen Stellenwert einnimmt, wurden jedoch keine speziellen Verwaltungsmöglichkeiten umgesetzt.

²<http://www.jelenajovanovic.net/LOCO-Analyst/>

4.2 Ontology-based Framework for Context-Aware Mobile Learning

Jawad Berri et al. [4] wählen in ihrem Framework einen etwas anderen Weg, um Lerninhalte zu erstellen, auszuwählen und anschließend an den Benutzer zu übermitteln. Die Grundlage für ihr ontologiebasiertes Framework wird aus einer regelbasierten Ontologie gebildet, welche die Anpassung der Lerninhalte an den Anwender ermöglicht. Als zweite Hauptkomponente dient ein Suchagent, der in weiterer Folge für die eigentliche Auswahl von Lernobjekten zuständig ist [4].

Grundsätzlich wird das Themengebiet, welches den Lernenden durch die mobile Lernapplikation nähergebracht werden soll, in einer Ontologie strukturiert (vergleichbar mit der *Domain Ontology* bei *m-LOCO*). Anschließend erstellt das System unter Berücksichtigung des Profils, welches für jeden Lernenden existiert, eine kompaktere, kontextbezogene Ontologie. Für diesen Vorgang kommt eine Wissensbasis zum Einsatz, welche eine Reihe von Regeln enthält, die den Inhalt, basierend auf Vorgaben des Lernenden und dessen Profil, einschränken lassen. Diese kontextbezogene Ontologie dient in weiterer Folge als Ausgangsbasis für den Suchagenten, der passende Lernobjekte aus verschiedenen Quellen im Web auswählt und daraus ein sogenanntes *learning web*³ erstellt. In einem letzten Schritt wird sichergestellt, dass die im *learning web* enthaltene Sequenz von Lernobjekten innerhalb der zeitlichen Einschränkung des Lernenden konsumiert werden können. Der gesamte Ablauf ist ebenfalls in Abbildung 4.2 ersichtlich. Die Schritte sind aus dem Grund als Zyklus dargestellt, weil eventuell erneut begonnen werden muss, wenn zeitliche oder profilbezogene Einschränkungen nicht eingehalten werden konnten [4].

Die größte Stärke dieses Systems ist der hohe Automatismus bei der Erstellung des *learning web*, welches anschließend dem Benutzer übermittelt wird. Dabei werden Kontextinformationen wie das Profil des Lernenden und zeitliche Einschränkungen berücksichtigt, womit genau die richtige Menge an relevanten Informationen in einer passenden Reihenfolge übermittelt wird.

4.3 Context-Aware Mobile Learning with a Semantic Service-Oriented Infrastructure

Ebenfalls ein ontologiebasiertes Framework wird von Chia et al. [6] vorgestellt, wobei der Schwerpunkt auf eine serviceorientierte Architektur gelegt wurde, welche aus einer Komponente zur Wissensverwaltung und einem Abfragemodul besteht. Diese lose Kopplung führt zu gekapselten, austauschbaren Modulen, was beispielsweise die Integration von Inhalten, welche andere

³Das *learning web* enthält eine Folge von Lernobjekten, welche dem Lernenden in einer bestimmten Reihenfolge präsentiert werden.

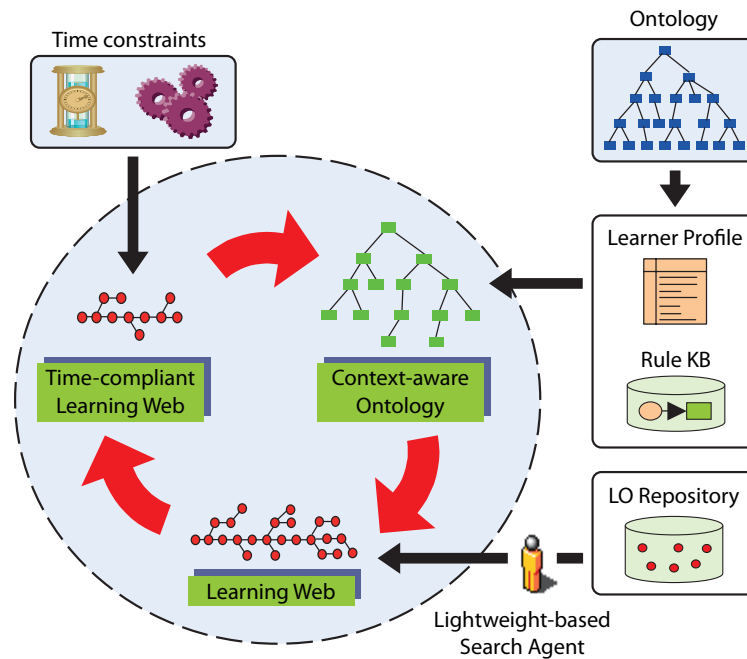


Abbildung 4.2: Der Ontologie-Mapping Zyklus [4].

Anbieter zur Verfügung stellen, ermöglicht [6].

Die übergeordnete Grundstruktur dieses Systems, das von den Autoren *KnowleMobiLe* genannt wird, kann in Abbildung 4.3 eingesehen werden. Ein als *Knowle-driven Content Layer* bezeichnetes Subsystem beinhaltet die Komponenten zur Wissensverwaltung und Abfrage. Ebenfalls darin enthalten ist ein System zur Extraktion von Inhalten aus Dokumenten, welches automatisch Domain-Ontologien erstellt und befüllt, wozu beispielsweise Webseiten, relationale Datenbanken oder Textsammlungen als Informationsquellen herangezogen werden. Die übergeordnete Ontologie wird ebenfalls automatisch generiert und dazu verwendet, semantische Verbindungen zwischen den einzelnen Domain-Ontologien herzustellen. Der Abfrage und Schlussfolgerungsmechanismus ermöglicht das logische Ableiten von zusätzlichen Informationen und überprüft gleichzeitig die Daten auf mögliche Inkonsistenzen. Zusätzlich können mit dessen Hilfe Abfragen auf die Datenbasis ausgeführt werden, um beispielsweise alle zu einem spezifischen Thema relevanten Objekte auszuwählen [6].

Für mobile Endgeräte stellt das Subsystem *Mobile Event Listeners and Actions* clientseitig eine multimodale Schnittstelle zur Verfügung, welche den Anwender in der Wahl des Kommunikationsweges möglichst wenig einschränkt und so beispielsweise SMS, MMS oder auch ein Sprachinterface erlaubt. Das System erkennt anhand spezieller Listener für das jeweilige Kom-

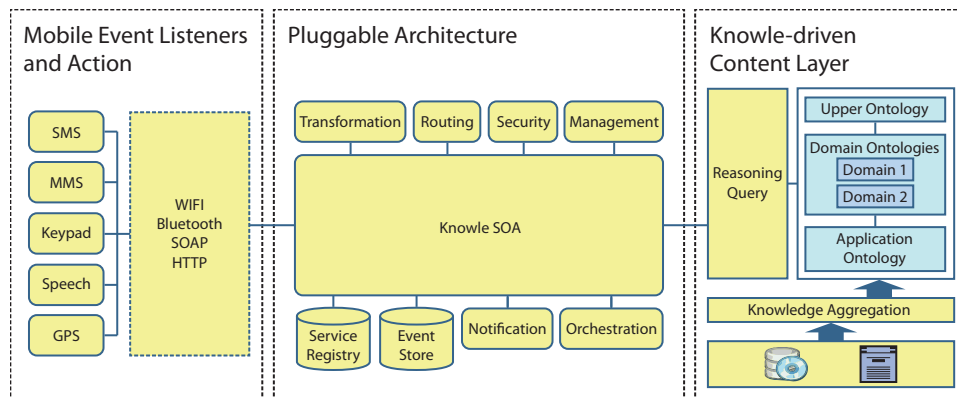


Abbildung 4.3: KnowleMobileLe Architektur [6].

munikationsmittel die Anfrage und schickt in weiterer Folge die Ergebnisse wieder an den Absender. Auch Kommunikation ohne direkte Eingaben des Benutzers ist möglich, um zum Beispiel die GPS-bezogene Kontextinformationen zu aktualisieren [6].

Zwischen den beiden bereits erläuterten Subsystemen steht ein Enterprise Service Bus (ESB), welcher in der Grafik als *Pluggable Architecture* ausgewiesen ist. Dieser stellt eine lose gekoppelte Verbindung zwischen mobilem Gerät und der Wissensbasis dar, wodurch einerseits die Subsysteme klar auseinander gehalten werden können, andererseits die Rollen des Ontologieverantwortlichen und des Lernenden auch systemseitig eindeutig abgegrenzt werden. Darüber hinaus ist dieses Subsystem auch für administrative Aufgaben zuständig, wie beispielsweise das Sicherheitsmanagement [6].

Dieser Ansatz führt zu einer für den Anwender sehr pragmatischen Lösung, die es ihm erlaubt, durch ihm bereits bekannte Interfaces mit dem System zu interagieren, wobei auch ein in Java entwickelter Client zur Verfügung steht. Aufseiten der Inhalte ermöglicht die lose gekoppelte Architektur, Daten aus einer Vielzahl von unterschiedlichen Informationsquellen zu extrahieren und daraus eine konsistente Wissensbasis zu generieren, welche dem Anwender für Abfragen zur Verfügung steht.

Kapitel 5

Eigener Ansatz

Basierend auf den in vorhergehenden Kapiteln dargestellten Grundlagen wird an dieser Stelle ein Ansatz vorgestellt, der kontextabhängiges Mobile Learning unter Berücksichtigung theoretischer Rahmenbedingungen ermöglicht, um Potenziale des mobilen Lernens auszuschöpfen. Grundsätzlich besteht das erstellte System aus einem Server, welcher für die Verwaltung der Kontextinformationen und Entitäten zuständig ist und einem Client, der die benötigten Kontextinformationen über vorhandene Sensoren ermitteln und zur Verfügung stellen kann. Zusätzlich bietet der Client Funktionalitäten, durch welche der Lernende mit dem System interagieren und Lernobjekte konsumieren kann.

Das erstellte Framework ist für viele Anwendungsfälle einsetzbar, wobei der Schwerpunkt auf einer kompakten und leicht erweiterbaren Lösung liegt. Im folgenden Abschnitt werden Anforderungen sowie einzelne Systembestandteile basierend auf den theoretischen Grundlagen beschrieben. Informationen zur konkreten Implementierung sind hingegen in Kapitel 6 zu finden.

5.1 Anforderungen

Das System ist dafür zuständig, speziellen Anforderungen gerecht zu werden, welche sich einerseits aus Grundlagen des mobilen Lernens ergeben, andererseits das Anwendungsgebiet etwas näher spezifizieren, wie beispielsweise die hohe Priorisierung der Ortsbezogenheit.

Ortsbezogenheit: Ein spezielles Anwendungsgebiet des umgesetzten Systems liegt in der Unterstützung ortsbezogener, mobiler Lernanwendungen. Inhalte können in Verbindung mit Standorten gesetzt werden und es erfolgt genau dann eine Übermittlung an die Lernenden, wenn sie sich in deren Nähe aufhalten. Dazu muss die Position der Lernenden am mobilen Gerät laufend abgefragt werden, um den Server davon in Kenntnis zu setzen, welcher ba-

sierend auf dem Standort und anderen Kontextinformationen die verfügbaren Lernobjekte ermittelt. Damit wird eine Grundlage für situiertes Lernen hergestellt, wodurch die Lernerfahrung dort gemacht werden kann, wo ein direkter Zusammenhang zu den Lerninhalten besteht. Zudem kann über die Position ein Bezug zur jeweiligen Umgebung des Lerngegenstands hergestellt werden, was als Voraussetzung für Lernen im authentischen Kontext dient.

Strukturierungsmöglichkeiten: Die Lerninhalte können so aufbereitet werden, dass die Möglichkeit besteht, sie strukturiert in kleinen Lernobjekten an die mobilen Endgeräte zu übertragen. Dies minimiert einerseits die jeweils übertragene Datenmenge, andererseits lassen größenbezogene Einschränkungen der Anzeige ohnehin nur die Darstellung eines beschränkten, gut aufbereiteten Informationsvolumens zu. Die Kapselung in kleinere Objekte vermindert somit Probleme, welche durch die relativ kleinen Displays der mobilen Endgeräte entstehen und ermöglicht es den Lernenden, die kleineren, abgegrenzten Informationseinheiten auch mit geminderter Konzentrationsfähigkeit zu verarbeiten. Zusätzlich ist es möglich, Lernobjekte und Gruppen von Lernobjekten so anzuordnen, dass sie nur verfügbar werden, wenn alle vorhergehenden Lernobjekte bereits konsumiert wurden.

Interessen: Bei der Auswahl der Lernobjekte spielt als Kontextinformation über die Lernenden nicht nur das Vorwissen in Form bereits abgerufener Lernobjekte eine Rolle, sondern auch deren Interessen. Eine Berücksichtigung dieser Informationen erlaubt eine systemseitige Personalisierung des Lernprozesses, wodurch die Lernerfahrung für die Anwender individualisiert werden kann. Dazu müssen Lerninhalte diversen Interessensgebieten zugeordnet und gleichzeitig die Vorlieben der Lernenden verfolgt werden. Diese Informationen dienen ebenfalls einer gezielteren Auswahl der einzelnen Lernobjekte, weshalb sie in der jeweiligen Abfrage entsprechend berücksichtigt werden müssen.

Automatisierung: Ziel des Systems ist eine möglichst optimale Auswahl der Lernobjekte in Abhängigkeit von zur Verfügung stehenden Kontextinformationen, wobei der Prozess automatisiert ablaufen soll. Dies vermindert den Aufwand auf Seite der Lernenden, um die mentalen Ressourcen der Anwender für das Verarbeiten der Lerninhalte aufzusparen. Dennoch werden im Sinne des individuellen Lernens Möglichkeiten zur eigenständigen Exploration der Lerninhalte angeboten, um selbstständigen Wissensaufbau zu unterstützen.

<i>Interaktion</i>	<i>Beschreibung</i>
Aktuelles Lernobjekt	Dieses Interface ermöglicht es dem Lernenden, das aktuell ausgewählte Lernobjekt zu studieren und durch dessen Inhalte zu navigieren. Details zur Strukturierung der einzelnen Lernobjekte können in Abschnitt 5.3.4 eingesehen werden.
Historie	Bereits konsumierte Inhalte müssen weiterhin verfügbar bleiben, weshalb der Client eine Historie der Lernobjekte zur Verfügung stellt, auf welche bereits zugegriffen wurde.
Übersichtskarte	Die Anforderung, ortsabhängige Lernobjekte zu unterstützen, stellt eine sinnvolle Einsatzmöglichkeit für eine Übersichtskarte dar. Diese unterstützt den Lernenden bei der Orientierung und bietet gleichzeitig eine Zugriffsmöglichkeit auf ortsabhängige Lernobjekte, ohne dass sich der Anwender zu deren Position bewegen muss.
Verfügbare Lernobjekte	Nicht nur bereits konsumierte und ortsabhängige Lernobjekte sollen auswählbar sein, sondern alle verfügbaren Inhalte, für welche der Anwender die notwendigen Voraussetzungen erfüllt.
Einstellungen	Für die Auswahl der Inhalte sind die Interessen der Lernenden wichtig, weshalb sie möglichst automatisch verfolgt und aktualisiert werden müssen. Dennoch wird dem Benutzer die Möglichkeit gegeben, diese bei Bedarf zu adjustieren.

Tabelle 5.1: Interaktionsmöglichkeiten.

5.2 Interaktionsmöglichkeiten

Der Client stellt die primäre Schnittstelle des Systems für den Benutzer dar und ist eine wesentliche Komponente des Mobile Learning Frameworks. Zur Interaktion mit dem System stellt der Client dem Benutzer verschiedene Funktionalitäten zur Verfügung, welche auf der einen Seite ein möglichst simples Interface zur Darstellung der automatisch ausgewählten Informationseinheiten ermöglichen, andererseits benutzergesteuertes Explorieren und Anpassen der Einstellungen erlauben. In Tabelle 5.1 sind die grundsätzlichen Interaktionsmöglichkeiten, welche vom Client aus Anwendersicht zur Verfügung stehen, dargestellt.

Eine beispielhafte Ansicht des Clients, durch welche der Grundaufbau des grafischen Interfaces veranschaulicht wird, ist in Abbildung 5.1 dargestellt. Die Grafik zeigt die Hauptansicht des Programmes, wobei über die

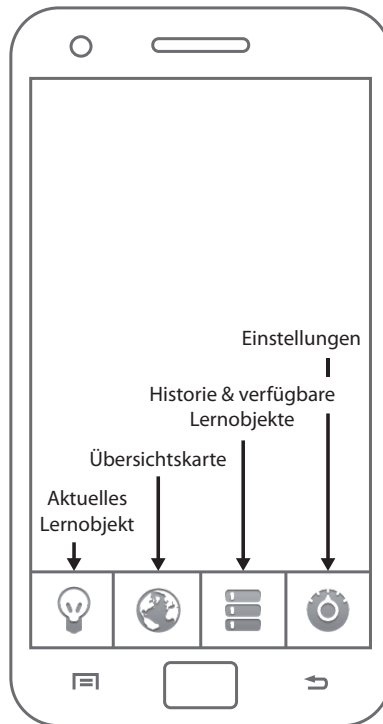


Abbildung 5.1: Grundsätzlicher Bildschirmaufbau des Clients.

einzelnen Karteireiter zu den Detailansichten navigiert werden kann. Die Benutzerschnittstelle ist bewusst sehr einfach gehalten und die Navigation mittels Karteireiter ermöglicht einen schnellen Zugriff auf alle Programmfunktionalitäten aus jeder Ansicht heraus. Die Programmelemente spiegeln die in Tabelle 5.1 angeführten Interaktionsmöglichkeiten wider, wobei die verfügbaren und bereits konsumierten Lernobjekte in einer Ansicht zusammengefasst sind. Auf die einzelnen Programmansichten im Detail wird in Abschnitt 6.2.3 eingegangen.

5.3 Aufbereitung der Entitäten und Kontextinformationen

Die Verwaltung und Berücksichtigung von Kontextinformationen ist eine essentielle Aufgabe von mobilen Lernanwendungen und kann, wie in Abschnitt 3.1 erläutert, auf verschiedene Arten umgesetzt werden. Aufgrund der Potenziale, die ontologiebasierte Systeme im Hinblick auf Schlussfolgerungsmechanismen, Informationsstrukturierung und Abfragemöglichkeiten bieten, dient ein solches System als Grundlage für die Datenbasis und anschließende Inhaltsauswahl. Das macht es notwendig, Kontextinformationen und Lern-

inhalte innerhalb einer Ontologie zu strukturieren.

In Abschnitt 2.6 wurden Individualität, Zeit, Standort, Aktivität und Beziehungen als relevante Kategorien von Kontextinformationen identifiziert, welche in kontextsensitiven Anwendungen eine Rolle spielen. Diesen Kategorien kommt auch in dem hier vorgestellten Ansatz eine große Bedeutung zu, sie werden in den erstellten Ontologien jedoch anders strukturiert.

Kontextinformationen werden im erstellten System nicht für beliebige, universell definierbare Entitäten verwaltet, sondern individuell angepasst für Lernende und Lernobjekte. Jegliche Entität, die als Lerninhalt dienen soll, kann jedoch abstrahiert als Lernobjekt abgebildet werden. Die formale Beschreibung von Lernenden und Lernobjekten ist jeweils in einer eigenen Ontologie definiert, welche auch den Individualitäts-Kontext abdeckt. Soziale, funktionale und kompositionale Beziehungen zwischen Lernobjekten und Lernenden sind ebenfalls direkt in diesen Ontologien abgebildet.

Der Aktivitätskontext, also die aktuellen Ziele des Lernenden, findet in diesem System serverseitig keine direkte Abbildung innerhalb der verwendeten Ontologien. Indirekt kann jedoch angenommen werden, dass Benutzer, welche den Client in Anspruch nehmen, ihr Wissen über die von der Lernanwendung behandelte Domäne erweitern möchten. Des Weiteren beinhaltet die *Lernenden*-Ontologie auch die Interessen der Lernenden, welche bei der Auswahl der Lernobjekte berücksichtigt werden. Dies reicht in Verbindung mit zusätzlichen Funktionalitäten des Clients, wie beispielsweise bei aktuell geöffnetem Lernobjekt kein weiteres anzufordern, aus, die aktuellen Aktivitäten beziehungsweise die Ziele der Lernenden zu berücksichtigen.

Die beiden bereits erwähnten Ontologien machen sich zusätzlich zwei andere, etwas weniger komplexe Ontologien zunutze, um zeit- und standortbezogene Kontextinformationen verwalten zu können. Die Auslagerung in eine jeweils eigene Ontologie gewährleistet dabei die Vermeidung von Redundanzen, weil Lernobjekte sowie Lernende einen Zeit- und Standortkontext besitzen. Abbildung 5.2 zeigt diese vier Ontologien (*Lernobjekt*, *Lernende*, *Zeit* und *Standort*), welche zusammengesetzt als Datenbasis des erstellten Frameworks dienen und in weiterer Folge näher erläutert werden.

Zur Erklärung der im nächsten Abschnitt abgebildeten Grafiken ist anzumerken, dass Klassen grundsätzlich als abgerundete Rechtecke dargestellt werden, die *rdfs:subClassOf*-Beziehung durch einen durchgehenden, violetten Pfeil abgebildet wird und andere, gestrichelte Pfeile die im Text näher erläuterten abstrakten Eigenschaften repräsentieren. Konkrete Eigenschaften sowie andere für die Erläuterung nicht essentielle Details sind in den Abbildungen aus Gründen der Übersichtlichkeit nicht angegeben.

5.3.1 Zeit

Diese sehr simple Ontologie stellt zwei Klassen inklusive zugehöriger Eigenschaften zur Verfügung, um zeitbezogene Informationen ausdrücken zu kön-

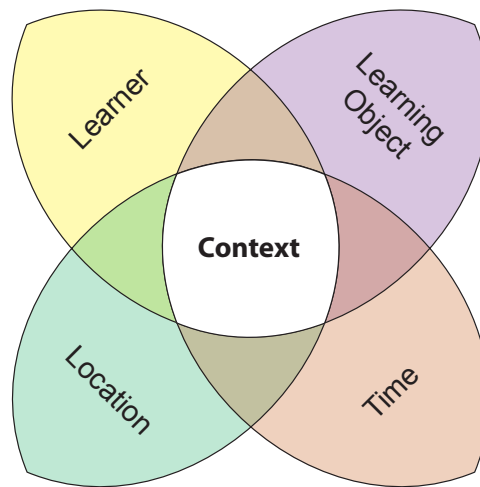


Abbildung 5.2: Grundsätzlicher Aufbau der Ontologie.

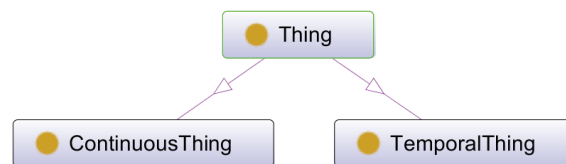


Abbildung 5.3: Struktur der Zeit-Ontologie.

nen. Abbildung 5.3 zeigt die relativ einfache Klassenstruktur dieser Ontologie. Individuen, welche mit einem gewissen Zeitpunkt in Verbindung gebracht werden müssen, können von der Klasse *TemporalThing* erben. Dadurch wird für sie die Eigenschaft *time* verfügbar, welche mit einem Zeitpunkt vom Datentyp *dateTime*¹ belegt werden kann. Die Klasse *ContinuousThing* steht hingegen solchen Individuen zur Verfügung, welche eine gewisse Zeitspanne besitzen. Eine Ableitung von dieser Klasse ermöglicht, dass bei Individuen die Eigenschaft *duration* mit einem *integer* belegt werden kann, welcher die Dauer in Sekunden ausdrückt. Die in der Grafik ersichtliche Klasse *Thing* wird vom Editor² vorgegeben und stellt jene Klasse dar, von der alle anderen abgeleitet werden müssen (vergleichbar mit *Object* in der objektorientierten Programmierung).

TemporalThing und *ContinuousThing* kommen in den anderen Ontologi-

¹Detaillierte Informationen zu den verwendeten Datentypen können unter <http://www.w3.org/TR/xmlschema-2/> abgerufen werden.

²Als Editor zur Gestaltung der in dieser Arbeit verwendeten Ontologien wurde Protégé verwendet. Nähere Informationen dazu können unter <http://protege.stanford.edu/> abgerufen werden. Die Abbildungen der Ontologien wurden mit dem von Protégé zur Verfügung gestellten OntoGraf Plugin erstellt.

en an verschiedenen Stellen zum Einsatz. Beispielsweise beinhalten manche Lernobjekte eine Audiodatei, deren Länge so auch in der Ontologie festgelegt und verwendet werden kann. Ebenso müssen Zugriffe von Lernenden auf Lernobjekte auch mit zeitlichem Zusammenhang gespeichert werden, wodurch ihre in der Ontologie verwaltete Repräsentation von der Klasse *TemporalThing* ableiten muss.

5.3.2 Standort

Um standortbezogene Informationen verwalten zu können, wird eine etwas umfangreichere Ontologie benötigt. Diese importiert zusätzlich das *Basic Geo Vocabulary*³, eine Ontologie, welche grundsätzliche Informationen wie Längengrad, Breitengrad, Höhe und geografische Punkte näher definiert. Auch die Klasse *Spatial Thing* stammt aus dieser Ontologie und definiert, dass alle abgeleiteten Klassen eine räumliche Ausdehnung oder Position besitzen. Um diese Ontologie besser mit Protégé verwalten zu können, wurde sie von RDF in OWL konvertiert.

In Abbildung 5.4 ist die grundsätzliche Struktur der Standort-Ontologie dargestellt, welche unter anderem *StationaryThing* wie auch *MovingThing* als Klassen einführt. Diese erlauben einerseits die Beschreibung von Entitäten, welche eine statische Position haben, andererseits können auch Entitäten abgebildet werden, die ihre Position verändern. Sich bewegenden Objekten vom Typ *MovingThing* ist es möglich, über die Eigenschaft *precedingPosition* mehrere *TemporaryPositions* anzugeben, wodurch der Bewegungsverlauf nachvollziehbar wird. Die Klassen *StationaryThing* und *TemporaryPosition* haben jeweils die Eigenschaften *Longitude* und *Latitude* des *Basic Geo Vocabularies*, um ihre jeweilige Position darstellen zu können. Zusätzlich erben *TemporaryPositions* auch die Eigenschaft *time* der Klasse *TemporalThing*, welche schon aus der Zeit-Ontologie bekannt ist, um den zeitlichen Verlauf von Positionsänderungen dokumentieren zu können.

Ein ideales Beispiel für Individuen des Typs *MovingThing* sind jene Anwender, welche sich während der Benutzung des Clients bewegen und deren Positionsverlauf sowie die aktuelle Position serverseitig verwaltet werden muss. Als *StationaryThing* sind hingegen ortsabhängige Lernobjekte deklariert, die sich an einer fixen Position befinden.

5.3.3 Lernende

Die Lernenden- und Lernobjekt-Ontologien sind im Aufbau komplexer als jene, die zeit- und standortbezogene Kontextinformationen modellieren. Die verwalteten Informationen sind etwas umfangreicher, es existieren Verweise auf die anderen Ontologien und sie gestalten auch den spezifischen Individualitäts- und Beziehungskontext der Lernenden beziehungsweise der Lern-

³Verfügbar unter <http://www.w3.org/2003/01/geo/>.

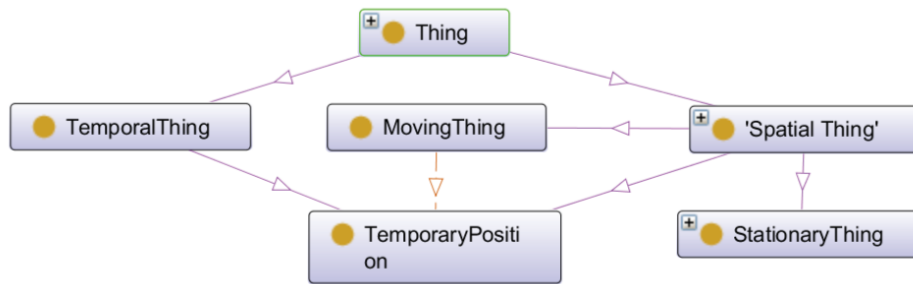


Abbildung 5.4: Struktur der Standort-Ontologie.

objekte. Zusammengefasst verwaltet die Lernenden-Ontologie aber folgende Informationen, welche sich auf Lernende beziehungsweise Benutzer beziehen:

- Personenbezogene Daten
- Beziehungen zwischen Personen
- Verwaltung der Interessen

Um grundsätzliches Wissen über Personen nicht neu strukturieren zu müssen, wird auch hier eine bestehende Ontologie eingebunden, die FOAF⁴-Ontologie. Diese ist für den hier vorgestellten Ansatz von Vorteil, weil sich durch ihre Verwendung bereits elementare Informationen über Personen modellieren lassen, wie beispielsweise Name, Geburtsdatum oder Benutzername. Zusätzlich werden dadurch die weiter unten angeführten Eigenschaften *topic* und *topic_interest* verfügbar, welche hilfreich für die Verwaltung der Interessen sind.

Abbildung 5.5 zeigt eine vereinfachte Struktur der Lernenden-Ontologie. Die zentrale Klasse *Person* aus der FOAF-Ontologie wurde neben den bestehenden, konkreten Eigenschaften vor allem um Interaktionsmöglichkeiten mit anderen Personen erweitert. Dadurch können die Benutzer mittels *Message* miteinander kommunizieren, wobei die Eigenschaften *author* und *recipient* angeben, wer in welcher Richtung Kontakt miteinander aufgenommen hat. Der eigentliche Inhalt der Nachrichten wird in Individuen der Klasse *Text* abgelegt, wobei die Eigenschaft *content* als Verbindung dient. Generell kann auch durch die in FOAF definierte Eigenschaft *knows* modelliert werden, welche Benutzer bereits auf beliebige Weise miteinander interagiert haben. Diese Eigenschaft ist in der Grafik durch die Referenz von *Person* auf sich selbst dargestellt.

Eine konkretere Ausprägung von *Person* stellt die Klasse *Learner* dar. Jedem *Learner* sind mehrere Themengebiete als *Topic* zugeordnet, wobei

⁴Die Friend of a Friend (FOAF) Ontologie ermöglicht es, Informationen zu Personen und Dokumenten, die von diesen erstellt wurden, abzubilden. Details sind unter <http://xmlns.com/foaf/spec/> zu finden.

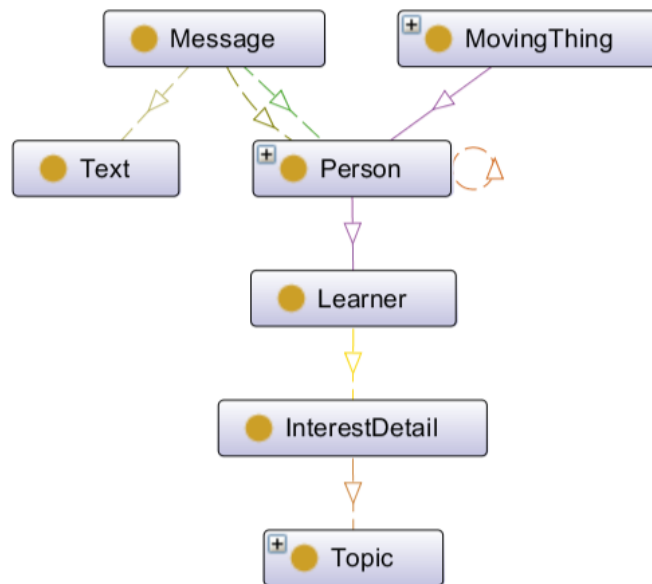


Abbildung 5.5: Struktur der Lernenden-Ontologie.

ebenfalls der Grad der Interessen des Benutzers an diesen Themen abgebildet werden kann. Dazu dient die Klasse *InterestDetail*, welche einerseits durch die Eigenschaft *topic_interest* von *Learner* mit diesem verbunden ist und andererseits den Verweis auf das zugehörige *Topic* mittels der gleich benannten Eigenschaft *topic* besitzt. Für jede mögliche Verbindung zwischen Lernenden und Themengebieten wird jeweils ein *InterestDetail* angelegt, welches die Stärke der jeweiligen Interessen als Gleitkommazahl verwaltet. Um nun eine interessensabhängige Lernobjekt-Auswahl unterstützen zu können, müssen auch die Lernobjekte den Themengebieten zugeordnet werden. Diese Verknüpfung und weitere Details zu den soeben erwähnten Lernobjekten sind im nächsten Abschnitt zu finden.

Anhand der Grafik ist auch erkennbar, welche Potenziale durch das Verknüpfen verschiedener Ontologien entstehen. Dadurch, dass jede Person von *MovingThing* aus der Standort-Ontologie abgeleitet wird, kann deren Positionsverlauf anhand von *TemporaryPositions* nachvollzogen werden. *TemporaryPositions* stammen wiederum von *TemporaryThing* ab, wodurch die exakten Zeitpunkte von Bewegungsänderungen oder auch die aktuellste Position ermittelt werden können. Setzt man letztere in Verbindung mit der Position von ortsabhängigen Lernobjekten, lassen sich bereits Informationen über das am nächsten liegende Lernobjekt anfordern. Dies ist nur ein Beispiel der Vielzahl an Möglichkeiten, die sich durch das Nutzen von Zusammenhängen aus den Ontologien ergeben.

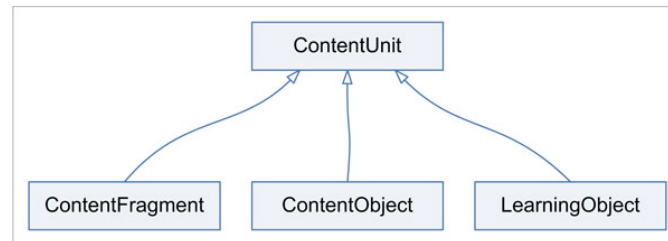


Abbildung 5.6: Die Hauptklassen der ALOCoM Core Ontology [34].

5.3.4 Lernobjekte

Die Lernobjekt-Ontologie besitzt eine höhere Komplexität als jene der Lernenden und wird dazu herangezogen, folgende Informationen strukturieren zu können:

- Aufbau und Inhalt der Lernobjekte
- Historie der Zugriffe durch Lernende
- Zuordnung zu Themengebieten

Die größte Herausforderung bei der Modellierung dieser Ontologie ist die Gestaltung der Lernobjekte, welche einerseits in kompakte, modulare Einheiten eingeteilt sein sollten, andererseits aber soweit miteinander in Verbindung stehen müssen, dass sinnvolle Lernsequenzen daraus gebildet werden können. Für die Lösung dieses Problems wird auf eine bereits existierende Ontologie zurückgegriffen, die *ALOCoM Core Ontology*.

ALOCoM Core Ontology

Die ALOCoM Core Ontology basiert auf dem Abstract Learning Object Content Model (ALOCoM)⁵ und ermöglicht eine Beschreibung von Lern- und Inhaltsobjekten sowie der Beziehungen zwischen ihnen. Abbildung 5.6 zeigt die wichtigsten Klassen sowie deren hierarchische Anordnung, wobei *ContentUnit* die übergeordnete Klasse darstellt, von der die anderen abgeleitet werden. Objekte vom Typ *ContentFragment* stellen die elementarsten Inhalte dar und bestehen für sich aus nur einem Medium wie Text, Bild, Audio oder Video. Ein *ContentObject* hingegen setzt sich aus *ContentFragments* oder wiederum *ContentObjects* zusammen, wodurch sich komplexere Inhalte aus verschiedenen Medientypen kombinieren lassen. Die technische Implementierung der Lernobjekte, welche in dieser Arbeit häufig erwähnt werden, entspricht genau diesen *ContentObjects*. Als übergeordnete Instanz dient die Klasse *LearningObject*, welche mehrere *ContentObjects* zu einem gemeinsamen Lernziel verknüpft.

⁵Weiterführende Informationen zu ALOCoM können unter <http://hmdb.cs.kuleuven.be/alocom/> gefunden werden.

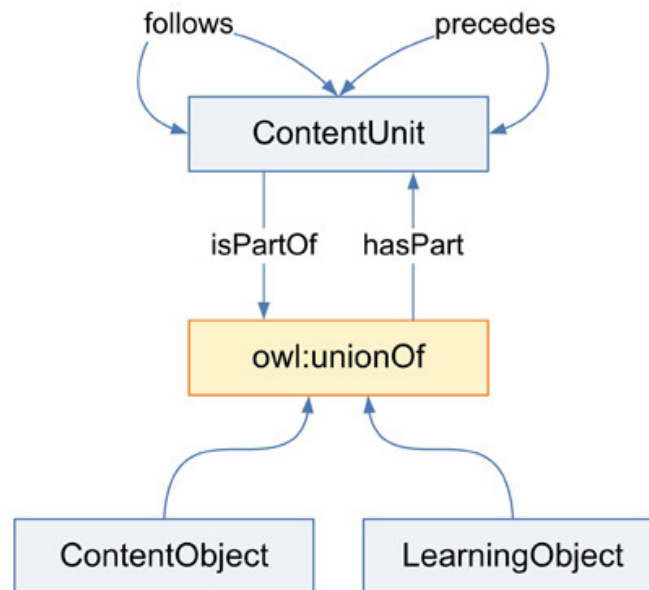


Abbildung 5.7: Beziehungen zwischen den Hauptklassen der ALOCoM Core Ontology [34].

Die Beziehungen zwischen den soeben erwähnten Klassen lassen sich anhand von Abbildung 5.7 genauer definieren. Durch die abstrakten, zueinander inversen Eigenschaften *follows* und *precedes* ist es möglich, *ContentUnits* in einer vorgegebenen Reihenfolge anzuordnen. Dies ist vor allem dann nützlich, wenn für nachfolgende *ContentUnits* Vorwissen benötigt wird, welches nur durch den Konsum vorgelagerter Inhalte erlangt werden kann. So könnte beispielsweise für eine Lernanwendung zum Thema Semantic Web definiert werden, dass erst dann Lernobjekte zum Thema Ontologien verfügbar werden, wenn der Lernende bereits Kenntnisse über Grundlagen des Semantic Web erlangt hat.

Für die interne Struktur der *ContentUnits* werden die ebenfalls abstrakten und zueinander inversen Eigenschaften *isPartOf* und *hasPart* eingesetzt. Sie erlauben das Festlegen der hierarchischen Struktur und geben Auskunft darüber, aus welchen *ContentObjects* ein *LearningObject* aufgebaut wird und welche *ContentFragments* wiederum zu einem *ContentObject* gehören. Dadurch, dass ein *ContentObject* aber auch andere *ContentObjects* und ein *LearningObject* andere *LearningObjects* beinhalten kann, sind eine hohe Flexibilität und beliebige hierarchische Gruppierungsmöglichkeiten gewährleistet. Nur *ContentFragments* ist es nicht erlaubt, aus mehr als einem elementaren Medientyp zu bestehen.

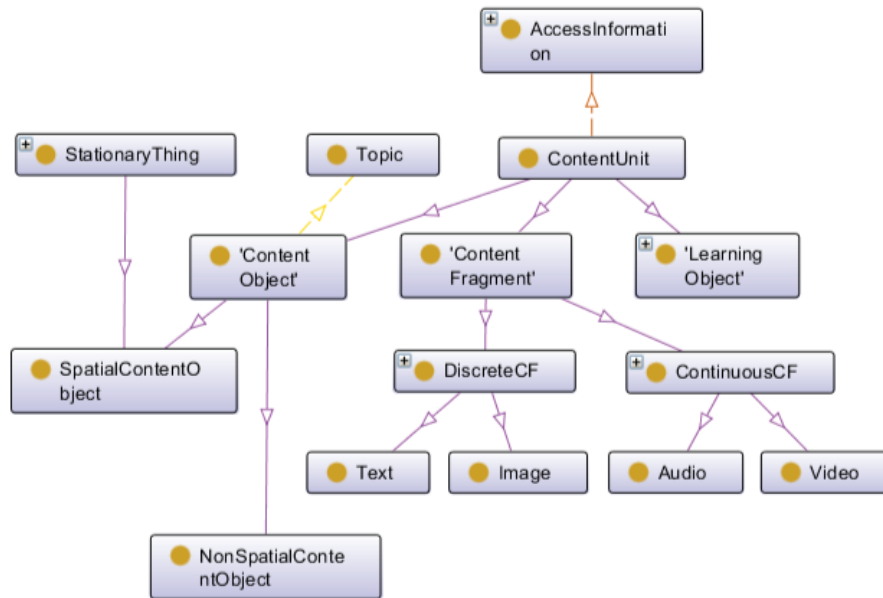


Abbildung 5.8: Struktur der Lernobjekt-Ontologie.

Erweiterung

Der Aufbau und Inhalt der Lernobjekte wird durch die ALOCoM Core Ontology bereits ausreichend abgedeckt. Um nun eine Historie der Zugriffe von Lernenden und eine Zuordnung der Lernobjekte zu Themengebieten systemseitig unterstützen zu können, definiert die Lernobjekt-Ontologie weitere Klassen und Eigenschaften, welche in Abbildung 5.8 eingesehen werden können. Zusätzlich ist die Untergliederung von *ContentFragment* in verschiedene Medientypen detaillierter erkennbar, wobei *Text* sowie *Image* von der Klasse *DiscreteCF* abgeleitet werden und *Audio* wie auch *Video* von *ContinuousCF* abstammen.

Um Inhaltsobjekte, die bereits abgerufen wurden, als solche zu kennzeichnen, wird *ContentUnits* bei jedem Zugriff eines Lernenden über die Eigenschaft *accessed* eine *AccessInformation* zugewiesen. Diese *AccessInformation* enthält einen Zeitstempel und einen Verweis auf *Learner*, um diese Informationen bei der Auswahl geeigneter Lernobjekte berücksichtigen zu können. Der Verweis auf *Learner* und die Tatsache, dass *AccessInformation* von *TemporalThing* ableitet sind in der Grafik nicht abgebildet, weil diese ohnehin schon sehr umfangreich ist.

Die Zuweisung zu Themengebieten ist eine weitere Information, welche von der Lernobjekt-Ontologie modelliert werden muss. Aus diesem Grund besitzt jedes *Learning Object* ebenfalls die abstrakte Eigenschaft *topic*⁶, wo-

⁶Die Eigenschaft *topic* ist bereits in der Lernenden-Ontologie vorhanden, weshalb ei-

durch diese Verknüpfung ermöglicht wird und auch Interessen bei der Auswahl von passenden Lernobjekten berücksichtigt werden können.

5.4 Auswahl der Lernobjekte

Informationen, welche durch die in Abschnitt 5.3 beschriebenen Ontologien verwaltet werden, können nun dazu genutzt werden, Lernobjekte kontextabhängig auszuwählen. Der hier vorgestellte Ansatz verwendet dazu die Abfragesprache SPARQL, welche eine gezielte Informationsselektion aus ontologiebasierten Datenbasen ermöglicht.

Grundsätzlich stößt der Client den Auswahlprozess an, indem dieser eine Aktualisierung der Kontextinformationen erkennt und den Server davon in Kenntnis setzt⁷. Anschließend wird serverseitig überprüft, ob basierend auf dem aktuellen Informationsstand der Ontologie Lernobjekte verfügbar sind. Ist dies der Fall, werden die ausgewählten Lernobjekte nach ihrer Relevanz gereiht und das passendste Lernobjekt an den Client übermittelt. Dieser Prozess ist essentiell für die mobile Lernanwendung und die konkrete SPARQL-Abfrage basiert auf folgenden Kontextinformationen:

- Position des Anwenders und ortsabhängiger Lernobjekte
- Zugriffshistorie des Anwenders auf Lernobjekte
- Interessen des Anwenders in Bezug auf Themengebiete der Lernobjekte

Wie bereits ausgeführt, ist der Auswahlprozess in mehrere Schritte geteilt, wobei die Vorauswahl aller aktuell geeigneten Lernobjekte und deren anschließende Reihung die wichtigsten Vorgänge darstellen. Ortsabhängige Kontextinformationen und dem System bekanntes Vorwissen in Form der Zugriffshistorie werden in der Vorauswahl berücksichtigt, während Interessen als Kriterium für die anschließende Reihung dienen.

5.4.1 Vorauswahl

Die im Folgenden vorgestellte Abfrage ist zur Vereinfachung der Beschreibung in mehrere Teile gegliedert, wobei die einzelnen Codeabschnitte zusammengeführt die gesamte SPARQL-Query ergeben. Sie wird im Framework zur Vorauswahl jener Lernobjekte verwendet, welche aufgrund des aktuellen Vorwissens und der Position des Anwenders in Frage kommen. Um diese relativ kompakte Abfrage zu ermöglichen, werden Schlussfolgerungsmechanismen und eine spezielle Filterfunktion verwendet, welche die reichhaltigen Möglichkeiten ontologiebasierter Informationsverwaltung sowie -selektion aufzeigen.

ne Unterscheidung durch verschiedene Namensräume notwendig ist. Die genaue Identifizierung der Eigenschaft *topic* findet somit mittels <http://xmlns.com/foaf/0.1/topic> beziehungsweise <http://www.fh-hagenberg.at/ontologies/learningObject.rdf#topic> statt.

⁷Details zur technischen Implementierung von Server, Client und deren Kommunikationsschnittstellen kann in Kapitel 6 gefunden werden.

Als Benutzer wird im Beispiel der Benutzer „John Doe“ angenommen, welcher über die URI `http://example.org#JohnDoe` eindeutig identifiziert wird.

Initialisierung:

Im ersten Codeabschnitt sind vor allem Namensräume sowie deren Präfixe definiert, welche daraufhin in der Abfrage verwendet werden können und eine bessere Lesbarkeit gewährleisten. Um Zugriff auf die benötigten Kontextinformationen zu haben, greift die Abfrage auf Elemente aus der Standort- (darunter auch das modifizierte Basic Geo Vocabulary), Zeit- sowie Lernobjekt-Ontologie (inklusive ALOCoM Core Ontology) zu, welche im vorangehenden Abschnitt näher definiert wurden. Der letzte festgelegte Namensraum (Präfix *cfn*) ermöglicht die Verwendung zusätzlich deklarierter Filter. Grundsätzlich liefert diese Abfrage URIs zurück, welche zur Identifizierung der ausgewählten *ContentObjects* eingesetzt werden.

```

1 PREFIX location:<http://www.fh-hagenberg.at/ontologies/locationContext.
  rdf#>
2 PREFIX geo:<http://www.fh-hagenberg.at/ontologies/wgs84pos0wl.rdf#>
3 PREFIX lo:<http://www.fh-hagenberg.at/ontologies/learningObject.rdf#>
4 PREFIX alocom-core:<http://jelenajovanovic.net/ontologies/loco/alocom-
  core/ns#>
5 PREFIX time:<http://www.fh-hagenberg.at/ontologies/timeContext.rdf#>
6 PREFIX ex:<http://example.org#>
7 PREFIX cfn:<http://www.fh-hagenberg.at/ontologies/custom-functions/>
8 SELECT ?contentObject WHERE {
9   ?contentObject a alocom-core:ContentObject .
10  FILTER NOT EXISTS {
11    ?contentObject lo:accessed ?accessInformation .
12    ?accessInformation lo:learner ex:JohnDoe }

```

Das erste Tripel des Graph-Musters der *WHERE*-Klausel gibt an, dass es sich bei Objekten, welche auf die Variable *?contentObject* gebunden werden können, um Objekte des Typs *alocom-core:ContentObject* handeln muss. Die Filterfunktion schließt darüber hinaus alle *ContentObjects* aus, auf welche der Benutzer *ex:JohnDoe* bereits zugegriffen hat, um diese nicht wiederholt an das mobile Endgerät dieses Anwenders zu übermitteln.

Position:

Der hier dargestellte Auszug definiert zusammengefasst, dass sich ortsabhängige Lernobjekte (vom Typ *lo:SpatialContentObject*) in der Nähe des Anwenders befinden müssen, um ausgewählt werden zu können.

```

1 FILTER NOT EXISTS {
2   ?contentObject a lo:SpatialContentObject
3   {
4     SELECT ?learnerLat ?learnerLong WHERE {
5       ex:JohnDoe location:precedingPosition ?latestPosition .
6       ?latestPosition time:time ?datetime .

```

```

7     ?latestPosition geo:lat ?learnerLat .
8     ?latestPosition geo:long ?learnerLong
9   }
10  ORDER BY DESC(?datetime)
11  LIMIT 1
12 }
13 FILTER NOT EXISTS {
14   ?contentObject geo:lat ?lat .
15   ?contentObject geo:long ?long .
16   ex:JohnDoe location:spatialInterestRange ?distance .
17   FILTER (cfn:haversine(?lat, ?long, ?learnerLat, ?learnerLong) < ?
distance) } }

```

Um die aktuellen Koordinaten des Benutzers zu ermitteln, wird eine Subquery⁸ eingesetzt. So wird die dem Zeitstempel nach aktuellste Position ausgewählt und die Variablen *?learnerLat* sowie *?learnerLong* mit deren Standortdaten belegt. Anschließend dienen diese Informationen dazu, nur solche ortsabhängigen Lernobjekte auszuwählen, welche innerhalb einer vom Benutzer einstellbaren Reichweite (*?distance*) liegen. Die Überprüfung erfolgt mithilfe der eigens dafür definierten Funktion *cfn:haversine*⁹, welche die Distanz zwischen Anwender und ortsabhängigem Lernobjekt berechnen lässt.

Vorwissen:

Als weitere wichtige Kontextinformation wird das Vorwissen des Benutzers in Form bereits konsumierter Lernobjekte durch die Abfrage berücksichtigt. Hier reicht es jedoch nicht aus, wenn nur auf die direkten Vorgänger des Lernobjektes zugegriffen wurde. Ist das Lernobjekt Teil einer übergeordneten *ContentUnit*, müssen auch deren vorgelagerte Inhaltsobjekte bereits abgerufen worden sein.

```

1  FILTER NOT EXISTS {
2   ?contentObject alocom-core:follows ?precedent .
3   FILTER NOT EXISTS {
4     ?precedent lo:accessed ?accessInformation .
5     ?accessInformation a lo:AccessInformation .
6     ?accessInformation lo:learner ex:JohnDoe } }
7  FILTER NOT EXISTS {
8   ?contentObject lo:isDerivedPartOf ?higherLevelContentUnit .
9   ?higherLevelContentUnit alocom-core:follows ?higherPrecedent .
10  FILTER NOT EXISTS {

```

⁸Subqueries sind Bestandteil der SPARQL 1.1 Spezifikation (siehe <http://www.w3.org/TR/2012/WD-sparql11-query-20120105/>, momentan noch Working Draft), werden aber vom verwendeten Resource Description Framework bereits unterstützt (siehe Abschnitt 6.1.1).

⁹Die Haversine-Funktion wird zur Berechnung von Entfernungen auf einer Kugel angewendet, was bei kleineren Entfernungen auf der Erdoberfläche nur in geringen Abweichungen zur realen Distanz resultiert und für diesen Anwendungsfall als ausreichend erachtet wird. Detailliertere Informationen können beispielsweise unter http://en.wikipedia.org/wiki/Haversine_formula gefunden werden.

```

11      ?higherPrecedent lo:accessed ?higherAccessInformation .
12      ?higherAccessInformation a lo:AccessInformation .
13      ?higherAccessInformation lo:learner ex:JohnDoe } } }

```

Die erste Filter-Funktion gibt an, dass kein *?contentObject* selektiert werden darf, welches Nachfolger eines anderen *ContentObjects* (angegeben durch *alocom-core:follows*) ist, auf das der aktuelle Lerner noch nicht zugegriffen hat. Dasselbe gilt für die zweite übergeordnete Filter-Funktion, nur dass hier die Überprüfung für alle übergeordneten *ContentUnits* (welche auf die Variable *?higherLevelContentUnit* gebunden werden) durchgeführt wird.

Eine besondere Bedeutung kommt der abstrakten Eigenschaft *lo:isDerivedPartOf* zu, welche bestehende Abhängigkeiten zu übergeordneten Lernobjekten und Lernzielen angibt. Diese Abhängigkeiten werden automatisch durch einen Reasoner¹⁰ erkannt und erstellt, wobei speziell definierte Schlussfolgerungs-Regeln zum Einsatz kommen. Eine dieser Regeln wird im Folgenden dargestellt (die Definition der Präfixe ist nicht darin enthalten):

```

1 Id: thesis_isDerivedPartOf1
2 a <alo:isPartOf> b
3 b <alo:isPartOf> c
4 a <rdf:type> <alo:ContentObject>
5 -----
6 a <lo:isDerivedPartOf> c

```

Durch die erste Zeile wird ein Bezeichner angegeben, welcher zur Identifizierung der Regel dient. Die darauf folgenden Zeilen bis zur Linie enthalten Vorbedingungen, welche erfüllt werden müssen, damit die nach der Linie angegebene Tripel zur Wissensbasis hinzugefügt werden. Diese Regel sagt aus, dass, wenn *a* vom Typ *alo:ContentObject* ist und durch die abstrakte Eigenschaft *alo:isPartOf* als Teil von *b* angegeben wurde und *b* wiederum Teil von *c* ist, auch gilt, dass *a* als Teil von *c* angenommen werden kann (angegeben durch die Eigenschaft *lo:isDerivedPartOf*).

Diese umfangreiche SPARQL-Abfrage berücksichtigt somit eine Vielzahl an Kontextinformationen und ermöglicht eine auf den Lernenden angepasste Vorauswahl der Lernobjekte. Diese werden anschließend nach Relevanz geordnet, wonach das passendste Lernobjekt an den Benutzer übermittelt wird.

5.4.2 Reihung

Die Reihung der vorausgewählten Lernobjekte ist etwas weniger komplex, weshalb an dieser Stelle die einzelnen Codeabschnitte nicht im Detail abgebildet sind. In einem ersten Schritt werden die URIs dieser Lernobjekte mithilfe einer weiteren SPARQL-Abfrage in Java-Objekte umgewandelt, welche unter anderem die eigentlichen Lerninhalte in der vordefinierten, sequentiellen

¹⁰Der OWLIM-Lite Reasoner wird von OWLIM-Lite zur Verfügung gestellt. Dieses RDF-Repository wird in Abschnitt 6.1.1 genauer vorgestellt.

Reihenfolge beinhalten. Darüber hinaus werden die den Lernobjekten zugeordneten Themengebiete aus der Ontologie ausgelesen, welche eine wichtige Rolle in der im Anschluss durchgeführten Reihung spielen.

Für die Berechnung der Rangfolge werden Lernobjekte danach gewichtet, wie sehr die jeweiligen Anwender an deren zugeordneten Themengebieten interessiert sind. Dieser Wert wird in der Datenbasis durch Instanzen der Klasse *InterestDetail* repräsentiert, welche für jede Verbindung zwischen Lernenden und Themengebieten existieren. Ist ein Lernobjekt mehreren Themen zugeordnet, wird das arithmetische Mittel der einzelnen Interessensausprägungen für den Vergleich herangezogen. Das Lernobjekt, welches den Interessen des Lernenden am ehesten entspricht, wird ausgewählt, als XML serialisiert und an den Client übermittelt. Dieser Kommunikationsvorgang zwischen Server und Client wird im nächsten Kapitel detailliert behandelt.

Kapitel 6

Umsetzung

Dieses Kapitel stellt die Beschreibung des entwickelten Frameworks dar, welches als mögliche Umsetzung von kontextsensitivem Mobile Learning mit ontologiebezogener Informationsverwaltung entwickelt wurde. Abbildung 6.1 veranschaulicht die wichtigsten Komponenten des Systems.

Grundsätzlich besteht die Systemarchitektur aus einem Client-Server-Modell, wobei der Server auf Anfragen des Clients mit komprimierten XML-Dateien antwortet, welche die angeforderten Informationen enthalten. Für die serverseitige, ontologiebasierte Datenbasis wird auf das Sesame-Framework

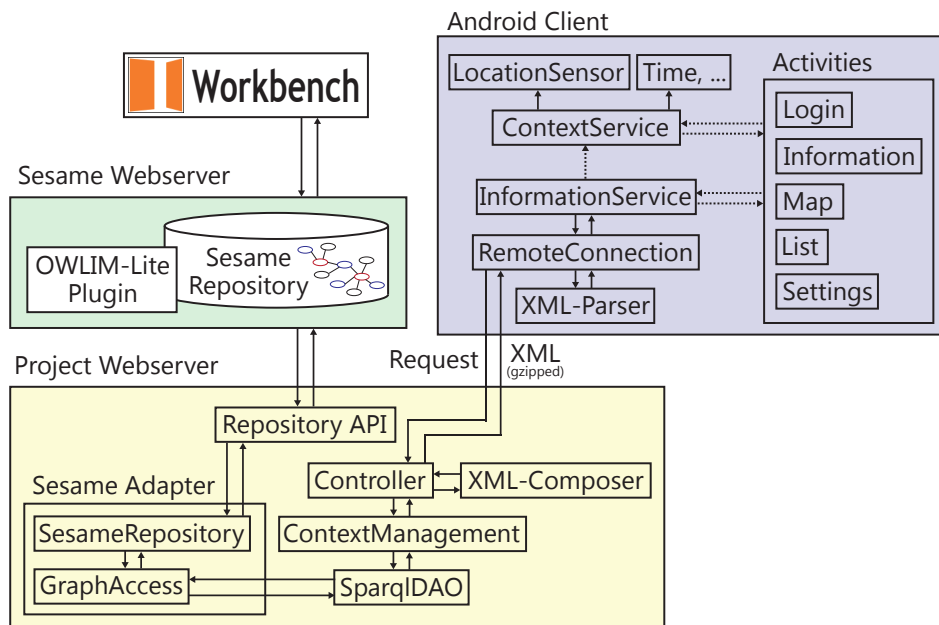


Abbildung 6.1: Systembeschreibung.

work zurückgegriffen. Als Client dient eine Android¹-Anwendung, welche unter Berücksichtigung der Grundlagen mobilen Lernens entwickelt wurde und die Möglichkeiten des Systems anhand einer funktionsfähigen Applikation für Endbenutzer aufzeigt. Details zu den einzelnen Systemkomponenten werden in folgendem Abschnitt, gruppiert in server- sowie clientseitige Bestandteile, dargestellt.

6.1 Server-Architektur

Der Server des Mobile Learning Frameworks wurde in Java entwickelt und ist für die Ausführung in einem Apache Tomcat 7 Servlet Container optimiert. Die ontologiebasierte Informationsverwaltung übernimmt ein RDF-Framework (Sesame), welches essentiell für die anderen Server-Komponenten ist und das neben einem Triplestore zur Datenhaltung auch Schnittstellen für die gezielte Informationsextraktion zur Verfügung stellt.

6.1.1 Sesame-Framework

Sesame ist ein Open-Source Java Framework und kann zum Speichern und Abrufen von RDF-Daten eingesetzt werden. Es ist ursprünglich als Forschungsprototyp für ein Projekt der Europäischen Union entstanden und wird derzeit als Community Projekt unter <http://openrdf.org/> weiterentwickelt. Sesame ist einfach zu installieren, bietet starke Performance und ist eines der führenden RDF-Frameworks [24].

Um Daten einpflegen, verwalten und abrufen zu können, stellt Sesame neben einer programmseitigen API² ebenso ein Web-Interface, die sogenannte Workbench, zur Verfügung, welche administrative Aufgaben vereinfachen und beschleunigen kann. Eine weitere Eigenschaft, die Sesame im Vergleich zu anderen Lösungen hervorhebt, ist die Möglichkeit, einzelne Komponenten austauschen oder erweitern zu können. Das wird im hier vorgestellten Mobile Learning Framework beispielsweise durch den Einsatz des OWLIM-Repositorys ausgenutzt.

Ebenfalls vorteilhaft für das Mobile Learning Framework ist die Aktualität der SPARQL-Unterstützung, die Sesame gewährleistet. Momentan vom W3C noch als Working Draft deklarierte Standards werden bereits zum Großteil unterstützt, wie beispielsweise die kurz in Abschnitt 5.4 erwähnte SPARQL 1.1 Spezifikation inklusive der im Framework verwendeten Subqueries. Darüber hinaus kann diese umfangreiche Funktionalität noch um individuelle Komponenten erweitert werden, zum Beispiel durch spezielle Funktionen, welche in *FILTER*-Bedingungen angewendet werden können.

¹Android ist ein weit verbreitetes Linux-basiertes Betriebssystem für mobile Geräte.

²Eine Programmierschnittstelle (application programming interface, API) wird von einem Softwaresystem zur Verfügung gestellt, um anderen Programmen auf Quellcode-Ebene Zugriff zu diversen Funktionalitäten zu ermöglichen.

OWLIM

OWLIM steht für eine Familie von semantischen Repositories, wobei für das hier verwendete Framework OWLIM-Lite zum Einsatz kommt. Wichtig für die weitere Arbeit ist eine genauere Definition des Begriffs *Semantic Repository*, den Bishop et al. [5, S. 33] in einer Publikation folgendermaßen abgrenzen:

There is no formal definition of the term 'semantic repository' so [...] we use this term for Database Management Systems (DBMS) that can be used to store, query and manage data structured according to the Resource Description Framework [23] (RDF) standard(s). Compared to Relational Database Management Systems (RDBMS), such systems use flexible ontological schemata where data is processed by an inference-engine according to a well-defined semantics.

Die OWLIM-Lite Edition (früher SwiftOWLIM) ist eine vorrangig im Hauptspeicher verwaltete RDF-Datenbank mit Unterstützung für Schlussfolgerungsmechanismen sowie Abfragen. Durch die Verwaltung im Hauptspeicher ist OWLIM-Lite sehr performant und kann dennoch mehrere Millionen Statements auf einem normalen Desktop PC verwalten [5]. Aus Persistenzgründen werden die Daten selbstverständlich auch auf Datenträgern abgelegt.

Im vorgestellten Mobile Learning Framework ist OWLIM-Lite notwendig, um Schlussfolgerungen auch auf OWL-Ebene zu ermöglichen. Sesame selbst unterstützt dies nur für RDF und RDFS. Darüber hinaus erlaubt OWLIM auch das Erstellen von eigenen Schlussfolgerungs-Regeln, welche beispielsweise für die Auswahl der Lernobjekte benötigt werden und in Abschnitt 5.4 dargestellt sind. Die Integration von OWLIM-Lite ist durch die einfache Austauschbarkeit von Komponenten in Sesame relativ einfach umsetzbar und so lassen sich alle Vorteile von Sesame und OWLIM-Lite nützen.

Alternativen

Sesame ist nur eines von zahlreichen Frameworks, welche die Verwaltung von RDF-Daten ermöglichen. Im Folgenden werden mehrere Alternativen aufgelistet, wobei sich diese Aufzählung an einer Aufstellung von Segaran et al. [24] orientiert:

Jena: Dieses Java Framework ist wie Sesame Open-Source und stellt ähnliche Funktionalität zur Verfügung, wie gut dokumentierte API-Schnittstellen, SPARQL-Unterstützung und Schlussfolgerungsmechanismen. Darüber hinaus kann OWL als Ontologiesprache ohne zusätzliche Plugins verwendet

werden. Jena hat im April 2012 den Status eines Apache Top Level Projekts erlangt³ (es war zuvor ein Apache Incubator Projekt), wodurch eine vielversprechende Grundlage für dessen weitere Zukunft geschaffen wurde. Weiterführende Informationen sind unter <http://jena.apache.org/index.html> verfügbar.

Redland: Das Redland-Framework umfasst eine Sammlung von mehreren modular aufgebauten C-Libraries, welche das Parsen, Speichern und Abfragen von RDF-Inhalten ermöglichen. Diese können in Projekten als einzelne Module oder als ganzes Framework verwendet werden, wobei API-Zugriffe aus verschiedenen Programmiersprachen (C#, Java, Perl, PHP, usw.) möglich sind. Die Projekthomepage von Redland befindet sich auf <http://librdf.org/>.

Oracle: Seit Version 11g bietet Oracle eine RDF-Management-Plattform an, welche im Gegensatz zu den bisher vorgestellten Frameworks als kommerzielle Lösung entwickelt wurde. Diese ermöglicht das Ablegen und Abrufen von RDF-Inhalten sowie automatisches Schlussfolgern, was auch bei OWL-Daten und mit benutzerdefinierten Regeln eingesetzt werden kann. Interessant bei dieser Lösung ist auch die Möglichkeit, ontologie-unterstützte Abfragen auf relationale Daten ausführen zu können. Detaillierte Informationen können unter http://www.oracle.com/technology/tech/semantic_technologies abgerufen werden.

Diese Aufzählung stellt nur einen kleinen Ausschnitt der verfügbaren RDF-Frameworks dar. Weitere Vertreter dieser Technologie sind beispielsweise Mulgara, OpenLink Virtuoso, Franz AllegroGraph sowie ARC.

Anbindung zu Sesame

Sesame bietet mittels API⁴ die Möglichkeit, auf lokal verwaltete Repositories zuzugreifen oder diese unabhängig vom eigentlichen Server zu betreiben. Für Letzteres ist der sogenannte *RemoteRepositoryManager* zuständig, der zum Erstellen, Löschen und Anfordern existierender Repositories verwendet werden kann. Einzelne Repositories werden durch die Klasse *Repository* abgebildet, welche neben dem Anfordern einer Verbindung (repräsentiert durch *RepositoryConnection*) auch das Initialisieren und Herunterfahren des jeweiligen Repositories ermöglichen. Als wichtigste Komponente kann jedoch die Klasse *RepositoryConnection* gesehen werden, welche neben dem eigentlichen Verbindungsauf- sowie abbau konkrete Verwaltungsaufgaben wie das Einpflegen, Abrufen und Löschen von Daten ermöglicht.

³<http://incubator.apache.org/projects/jena.html>

⁴Eine Dokumentation der API ist unter <http://www.openrdf.org/doc/sesame2/api/> verfügbar.

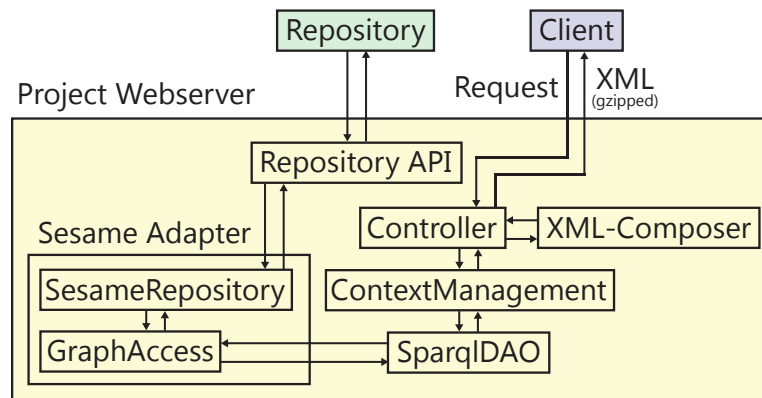


Abbildung 6.2: Vereinfachte Darstellung des Servers.

6.1.2 Aufbau

Abbildung 6.2 zeigt eine vereinfachte Darstellung des entwickelten Mobile Learning Framework Servers und dessen Schnittstellen nach außen. In weiterer Folge werden die einzelnen Komponenten näher erläutert, wobei aus Gründen der Übersichtlichkeit vorrangig auf die Architektur und weniger auf Implementierungsdetails eingegangen wird. Die *Repository-API* ist ein von Sesame zur Verfügung gestelltes Library, welches die bereits erwähnten Klassen der Sesame-API beinhaltet und die Kommunikation mit dem Repository ermöglicht.

Sesame Adapter

Grundsätzlich ist die Anbindung des Servers zum RDF-Framework modular aufgebaut, sodass mit relativ wenig Aufwand auf eine andere Lösung umgestellt werden kann. Dafür ist es notwendig, das verwendete Repository abgekapselt vom eigentlichen Mobile Learning Framework zu betreiben. Die Anbindung an dieses Repository erfolgt abstrahiert über das *IRepository*-Interface, wobei die zugrunde liegende Implementierung und der konkrete Verbindungsaufbau zum Sesame-Repository durch die Klasse *SesameRepository* übernommen werden. Diese ist auch für einen sauberen Verbindungsabbau zuständig.

Konkrete Operationen am Repository werden mithilfe der Klasse *GraphAccess* durchgeführt, welche Zugriff auf den im Triplestore verwalteten Graphen bietet. Beispiele für solche Methoden sind das Hinzufügen von Tripeln oder das Ausführen von SPARQL-Abfragen auf die enthaltenen Daten. *SesameRepository* und *GraphAccess* übernehmen somit die Kommunikation zur Sesame-API und stellen den anderen Server-Komponenten eine abstrahierte Schnittstelle zur Verfügung.

Controller

Der *Controller* ist für den Ablauf und die Steuerung zuständig, weshalb er Anfragen des Clients übernimmt und weiterverarbeitet. Zur Unterscheidung, welche Aktionen auf bestimmte Requests der Clients folgen, ist der Controller annotationsbasiert konfiguriert, was durch das verwendete Spring Web-MVC Framework⁵ ermöglicht wird. Abhängig vom Request entscheidet der Controller aber nicht nur, welche Handlungen ausgeführt werden müssen, er ist auch für das Auslesen, Überprüfen und Aufbereiten der Parameter zuständig, die beispielsweise den Benutzernamen oder die aktuelle Position des Anwenders beinhalten. Diese werden anschließend der Klasse *ContextManagement* übergeben, welche den Controller in weiterer Folge wieder mit Daten versorgt, der diese als XML verpackt, komprimiert und an den Client sendet.

ContextManagement

Die Klasse *ContextManagement* dient als zentrale Verwaltungsstelle für Kontextinformationen. Einerseits werden automatisch Änderungen festgestellt und an die Ontologie übermittelt, andererseits wird basierend auf diesen Informationen automatisch evaluiert, ob neue, relevante Lernobjekte verfügbar sind. Mithilfe der in Abschnitt 5.4 dargestellten Methoden wird, falls vorhanden, das aktuell passendste Lernobjekt ausgewählt und an den *Controller* für die Weiterverarbeitung übermittelt. Neben dieser Hauptaufgabe werden weitere Funktionalitäten zur Verfügung gestellt, wie das Erkennen aller in der Nähe befindlichen ortsabhängigen Lernobjekte, das Anpassen der Interessen oder das Ermitteln von Lernenden, zu denen der jeweilige Benutzer einen Bezug hat.

SparqlDAO

*SparqlDAO*⁶ ist eine Klasse, welche konkrete Methoden für den Zugriff auf die Ontologie und für deren Manipulation zur Verfügung stellt. Dazu implementiert diese Klasse die SPARQL-Queries, welche anschließend über den Sesame Adapter am Repository ausgeführt werden. Die zurückgelieferten Variablenbelegungen werden daraufhin in Java-Objekte umgewandelt, welche vom weiteren Framework verwendet und als XML-Dokumente serialisiert werden können. Des Weiteren stellt die Klasse *SparqlDAO* Schnittstellen zur

⁵Details können unter <http://static.springsource.org/spring/docs/2.5.x/reference/mvc.html> gefunden werden.

⁶DAO steht für Data Access Object und entspricht einem Entwurfsmuster, bei welchem der Zugriff auf die Datenquelle durch ein Interface abstrahiert wird, wodurch die Datenquelle im Bedarfsfall leicht ausgetauscht werden kann. Dies ist auch im beschriebenen Mobile Learning Framework der Fall.

Verfügung, welche das Hinzufügen, Bearbeiten und Löschen von Informationen ermöglichen.

XML-Composer

Wie bereits erwähnt, sind Daten, die vom Server an den Client gesendet werden, im XML-Format serialisiert. Dies übernimmt der *XML-Composer*, welcher die Document Object Model (DOM) Schnittstelle der *Java API for XML-Processing* (JAXP) verwendet, die ebenso zum Erstellen wie auch zum Parsen von XML-Dokumenten verwendet werden kann. Letzteres wird vom *XML-Parser* des Clients eingesetzt. Der *XML-Composer* kann beliebig verschachtelte Java-Objekte als XML serialisieren, solange diese eine bestimmte Struktur aufweisen. Dies ist insbesondere bei mehrfach verschachtelten Lernobjekten notwendig.

6.1.3 Schnittstellen für den Client

Der mobile Client nimmt mittels Http-Request Kontakt mit dem Server auf. Letztgenannter bietet dafür mehrere Schnittstellen an, die einerseits zum Anfordern neuer Informationen genutzt werden können, andererseits das Aktualisieren der ontologiebasierten Datenbasis ermöglichen. Die für das erstellte Framework zentralen Verbindungsmöglichkeiten werden in folgender Aufzählung genannt:

- **/login**: Identifizierung des Benutzers
- **/setContextInformation**: Aktualisieren der Kontextinformationen
- **/getNearbyEntities**: Abrufen nahe liegender Entitäten
- **/getAccessedLearningObjects**: Abrufen konsumierter Lernobjekte
- **/getAvailableLearningObjects**: Abrufen passender Lernobjekte
- **/getSpecificLearningObject**: Abrufen eines speziellen Lernobjektes
- **/getInterestDetails**: Abrufen der Interessen des Benutzers
- **/setInterestDetails**: Aktualisieren der Interessen des Benutzers
- **/getSpatialInterestRange**: Abrufen der zu prüfenden Reichweite
- **/setSpatialInterestRange**: Setzen der zu prüfenden Reichweite

Dem Client stehen damit mehrere Möglichkeiten zur Verfügung, Informationen zur Anzeige für den Benutzer anzufordern oder den Datenstand des Servers mit neuem Wissen anzureichern. Auch bei Requests, die als zentrale Funktion das Aktualisieren des Informationsbestandes am Server haben (wie **/setContextInformation**), wird die bereits hergestellte Verbindung genutzt um den Client über neu verfügbare Informationen wie Lernobjekte oder automatisch aktualisierte Interessensstärken zu benachrichtigen.

6.2 Client-Architektur

Der Client dient vorrangig dazu, die praktische Verwendbarkeit des ontologiebasierten Frameworks nachzuweisen und diese anhand einer Fallstudie (siehe Abschnitt 7.1) aufzeigen zu können. Die einzelnen Benutzerschnittstellen veranschaulichen einen großen Teil des Potenzials, welches durch die Verwaltung der Kontextinformationen in einer kompakten Ontologie mit reichhaltigen Schlussfolgerungs- und Abfragemöglichkeiten entsteht.

6.2.1 Zielplattform

Für die Entwicklung einer mobilen Lernanwendung bieten sich einige verschiedene Endgeräte (siehe Abschnitt 2.1.2) an, wobei Smartphones die größte Funktionsvielfalt in Verbindung mit einer hohen Verbreitung bieten und aus diesem Grund als Zielgeräte ausgewählt wurden. Im Speziellen wurde der Client für Smartphones mit Android-Betriebssystem entwickelt, das laut Gartner [32] im ersten Quartal 2012 mit 56,1 Prozent vor iOS (22,9 Prozent) den größten Marktanteil bei Smartphone-Verkäufen hatte. Der Client wurde als native Anwendung umgesetzt, um Vorteile wie einen einfachen Zugriff auf Sensoren, natives „Look and Feel“ und vom Betriebssystem bereitgestellte Android-Komponenten nutzen zu können.

6.2.2 Aufbau

In Abbildung 6.3 kann eine vereinfachte Darstellung der Architektur des Clients eingesehen werden. Die Kommunikation zum Server und die Interaktion mit den verfügbaren Sensoren wurde in Services ausgelagert, welche in Android-Systemen für Programmkomponenten benutzt werden, die länger andauernde Tätigkeiten ohne direkte Interaktion mit dem Benutzer im Hintergrund ausführen. Eine weitere, für den Client wichtige Art von Programmbestandteilen sind sogenannte Activities, die eine Bildschirmansicht implementieren, mit welcher der Benutzer interagieren und bestimmte zusammenhängende Aufgaben ausführen kann. Weitere Informationen zu Komponenten und generell zur Entwicklung mit Android können vor allem in den offiziellen API Guides⁷ gefunden werden.

ContextService

Das *ContextService* dient als zentrale Verwaltungsstelle für Sensoren. Änderungen der Kontextinformationen werden von den Sensoren registriert und direkt an dieses Service weitergegeben. Die anderen Komponenten der Applikation, welche sich für eine bestimmte Art von Kontextänderungen interessieren, werden daraufhin benachrichtigt und können aktualisierte Daten anfor-

⁷<http://developer.android.com/guide/components/index.html>

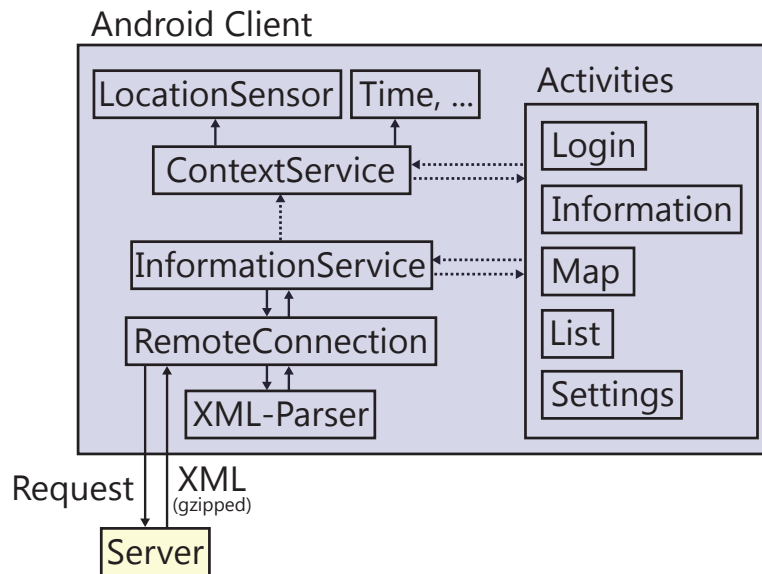


Abbildung 6.3: Vereinfachte Darstellung des Client.

dern. Ein Beispiel für eine derartige Komponente ist das *InformationService*, welches dem Server laufend aktuelle Kontextinformationen übermittelt.

InformationService

Die Kommunikation mit dem Server findet ausschließlich über das *InformationService* statt. Dieses entscheidet, wann Kontextinformationen an den Server übermittelt werden und es erhält darüber hinaus aktualisierte Informationen vom Server, sobald diese verfügbar sind. Beispiele für solche Objekte sind:

- Das aktuell am besten geeignete Lernobjekt (falls verfügbar)
- Ortsabhängige Lernobjekte und Lernende innerhalb eines bestimmten Radius
- Alle bisher konsumierten Lernobjekte
- Angepasste Interessen des Lernenden an Themengebieten

Andere Programmkomponenten, wie die Benutzerschnittstellen, werden über neu verfügbare oder geänderte Informationen benachrichtigt, woraufhin sie diese vom *InformationService* anfordern können. Für die eigentliche Verbindung zum Server ist die Klasse *RemoteConnection* zuständig.

RemoteConnection

Die Klasse *RemoteConnection* erlaubt es, Verbindungen zum Server aufzubauen. Eventuell vom Server übertragene Objekte werden in dieser Pro-

grammkomponenten empfangen, entpackt und mithilfe des *XML-Parsers* in Java-Objekte umgewandelt, welche wiederum vom *InformationService* verarbeitet und weitergeleitet werden können.

XML-Parser

Der *XML-Parser* entspricht dem Gegenstück zum *XML-Composer*. Er deserialisiert die vom Server im XML-Format erhaltenen Daten und wandelt sie wiederum in Java-Objekte um, was, wie auch beim Server, durch die Verwendung der *Java API for XML-Processing* ermöglicht wird. Die Erkennung, um welche Objekte es sich handelt, erfolgt über den Namen der jeweiligen Knoten, die beliebig verschachtelt sein können.

6.2.3 Benutzerschnittstellen

Neben den bereits vorgestellten Programmkomponenten besitzt der Client mehrere Benutzerschnittstellen, welche den Anwendern Interaktionsmöglichkeiten zur Verfügung stellen. Diese entsprechen in Android-Systemen sogenannten Activities und sind im Mobile Learning Framework in *Login*, *Information*, *Map*, *List* und *Setting* eingeteilt, wie durch Abbildung 6.3 ersichtlich wird. Der grundsätzliche visuelle Aufbau des Clients kann in Abschnitt 5.2 eingesehen werden, während auf die einzelnen Benutzerschnittstellen im folgenden Teil eingegangen wird.

Identifizierung

Die Login-Activity ist sehr überschaubar und bietet den Benutzern lediglich die Möglichkeit, sich bei existierendem Benutzer einzuloggen oder andernfalls einen Registrierungs-Vorgang einzuleiten. Diese Identifizierung der Anwender ist für den weiteren Programmverlauf unumgänglich, um die serverseitig verwalteten Daten eindeutig dem entsprechenden Benutzer zuweisen zu können.

Darstellung der Informationen

Zur Darstellung von Informationen wird die *Information-Activity* eingesetzt, deren visuelle Repräsentation in Abbildung 6.4 dargestellt ist. Der primäre Anwendungszweck besteht in der Anzeige von Lernobjekten und der Interaktion mit denselben. Lernobjekte bestehen, wie in Abschnitt 5.3.4 beschrieben, aus mehreren Teilen, zwischen denen mithilfe der Schaltflächen navigiert werden kann. Neben Text und Bild kann der Client auch Audiobestandteile von Lernobjekten wiedergeben. Die Informationsansicht dient zusätzlich zur Darstellung von erhaltenen Nachrichten, welche von anderen Benutzern übermittelt wurden. Ebenso kann der Anwender in dieser Ansicht selbst Nachrichten erstellen, die an andere Benutzer versendet werden können.

Für die Bereitstellung von Daten ist das *InformationService* zuständig. Darüber hinaus werden auch relevante Daten über Handlungen des Benutzers an den Server weitergeleitet, wie beispielsweise ein frühzeitiges Ausblenden von Lernobjekten, ohne dass eine Betrachtung der einzelnen Lernobjekt-Teile stattgefunden hat.

Übersichtskarte

Die Übersichtskarte dient der räumlichen Orientierung des Benutzers und ist in Abbildung 6.5 ersichtlich. Sie entspricht der grafischen Darstellung der *Map-Activity* und stellt ortsabhängige Lernobjekte sowie andere Benutzer in der näheren Umgebung auf einer Landkarte dar, die laufend durch Informationen vom *InformationService* aktualisiert wird. Des Weiteren sind vorhandene Karten-Einblendungen interaktiv und es können Details zu den in dieser Form dargestellten Lernobjekten angefordert werden. Ebenfalls ist es möglich, Lernenden, die auf dieser Karte abgebildet sind, mithilfe der Karten-Einblendung eine Nachricht zu schicken. Zur Darstellung der angeforderten Lernobjekte wie auch zur Erstellung der Nachricht wird aber die *Information-Activity* eingesetzt.

Zur Darstellung der Karte wird die Google Maps Android API⁸ eingesetzt. Diese Bibliothek ermöglicht mithilfe vorgefertigter Komponenten die Verwendung des Google Maps Dienstes auf Android Geräten.

Lernobjekt-Liste

Die *List-Activity* ermöglicht eine listenartige Darstellung von Lernobjekten. Dabei unterscheidet die Benutzerschnittstelle zwischen zwei unterschiedlichen Ansichten: Einerseits einer Liste der bereits abgerufenen Lernobjekte, andererseits einer Aufstellung von Lernobjekten, die im Hinblick auf aktuell verfügbare Kontextinformationen vom Server als momentan relevant eingestuft werden. Diese Benutzerschnittstelle ist in Abbildung 6.6 dargestellt.

Abgerufene Lernobjekte: Diese Liste stellt ein wichtiges Kriterium für den Lernfortschritt des Benutzers dar. Nachdem ein Anwender das momentan ausgewählte Lernobjekt konsumiert und ausgeblendet hat, sollte es in weiterer Folge für den Anwender abrufbar bleiben und nicht verschwinden. Diese Aufgabe erledigt die erste Ansicht, welche eine Liste bereits abgerufener Lernobjekte beinhaltet. Diese sind anhand der Themengebiete, denen sie zugeordnet sind, kategorisiert.

Verfügbare Lernobjekte: Ortsabhängige Lernobjekte können mithilfe der Übersichtskarte abgerufen werden. Diese Möglichkeit fehlt bei anderen,

⁸Details zur Google Maps Android API können unter <https://developers.google.com/maps/documentation/android/> abgerufen werden.



Abbildung 6.4: Visuelle Repräsentation der *Information-Activity*.

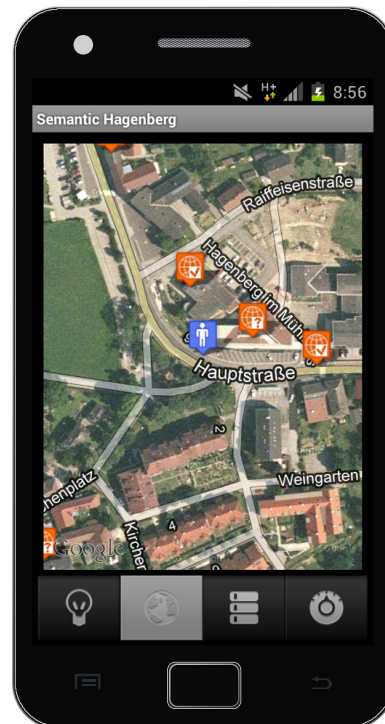


Abbildung 6.5: Visuelle Repräsentation der *Map-Activity*.

noch nicht vom Server übermittelten Lernobjekten, weshalb die *List-Activity* eine weitere Ansicht beinhaltet. Die Liste der verfügbaren Lernobjekte gibt dem Benutzer die Freiheit, selbst für ihn interessante Lernobjekte auszuwählen, sofern diese vom Server als passend für die momentan aktuellen Kontextinformationen eingestuft werden.

Einstellungen

Die *Settings-Activity* ist eine weitere Benutzerschnittstelle und ermöglicht den Anwendern das Anzeigen und Anpassen ihrer jeweiligen Interessen an verschiedenen Themengebieten. In Abhängigkeit von der Zugriffshistorie auf Lernobjekte der jeweiligen Benutzer werden deren Interessen vom System weitgehend automatisch adjustiert, weshalb diese Ansicht den Lernenden Einblick in die aktuelle Gewichtung von Interessen gibt. Darüber hinaus werden die Anwender hier dazu befähigt, die vom Server vorgegebene Einstufung selbst an ihre eigene Einschätzung anzupassen. Die grafische Darstellung dieser Benutzerschnittstelle ist in Abbildung 6.7 ersichtlich.

Zusätzlich zu den Interessen in einzelne Themengebiete lässt sich über

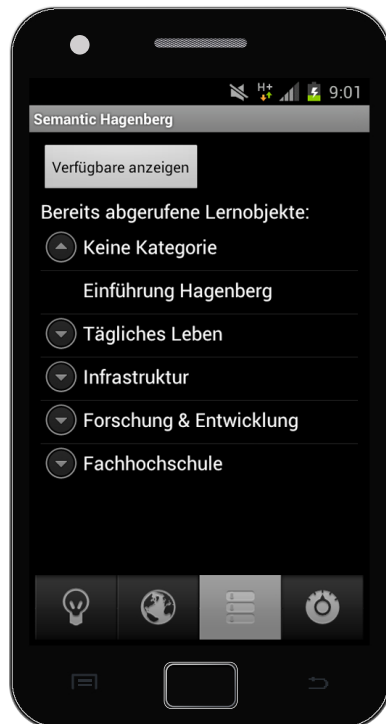


Abbildung 6.6: Visuelle Repräsentation der *List*-Activity.

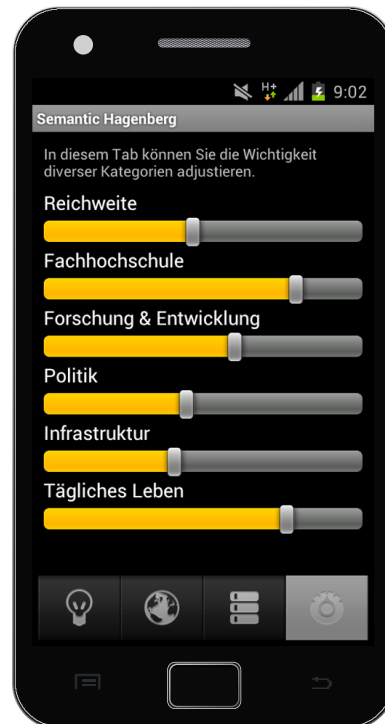


Abbildung 6.7: Visuelle Repräsentation der *Settings*-Activity.

die *Settings*-Activity auch die Entfernung adjustieren, ab welcher ortsabhängige Lernobjekte als Nahe wahrgenommen werden. Ortsabhängige Lernobjekte innerhalb dieser Reichweite werden in der serverseitigen Auswahl höher priorisiert.

Kapitel 7

Evaluierung

Das im Zuge dieser Arbeit vorgestellte ontologiebasierte Framework hat zum Ziel, kontextsensitives mobiles Lernen zu ermöglichen. Lernenden sollen die richtigen Informationen zum passenden Zeitpunkt vermittelt werden, wobei der Aufwand auf Seite der Lernenden minimiert wird. Aktuelle technische Möglichkeiten werden als Hilfestellung eingesetzt um die Grundlagen mobilen Lernens möglichst optimal umzusetzen und Anwender in ihrem Lernvorgang zu unterstützen. Dieses Kapitel hinterfragt das entwickelte System im Hinblick auf die Umsetzung der Prinzipien mobilen Lernens anhand einer kurzen Fallstudie.

7.1 Fallstudie

Um das erstellte Mobile Learning Framework anhand einer praxisnahen Problemstellung evaluieren und auf dessen Einsatzbarkeit überprüfen zu können, wurde das System mithilfe eines realitätsnahen Szenarios erprobt. Dazu wurde der Triplestore im Rahmen dieser Fallstudie mit Daten angereichert und die Funktionsfähigkeit des Frameworks durch die Verwendung von Android-Clients geprüft.

7.1.1 Szenario

Ein geeignetes Einsatzgebiet des Mobile Learning Frameworks besteht im Kennenlernen von Standorten wie großen Betriebsanlagen oder touristischen Zielen. Aus praktischen Gründen wurde als Örtlichkeit der Campus Hagenberg der FH Oberösterreich sowie dessen nähere Umgebung ausgewählt. Als Anwender werden Personen angenommen, die neu am Campus sind und diesen in relativ kurzer Zeit kennenlernen möchten, wie neue Studenten, Gastprofessoren oder Austauschstudenten.

Ohne die mobile Lernanwendung wird Wissen über Hagenberg vor allem durch zwischenmenschliche Kommunikation vermittelt oder anhand von

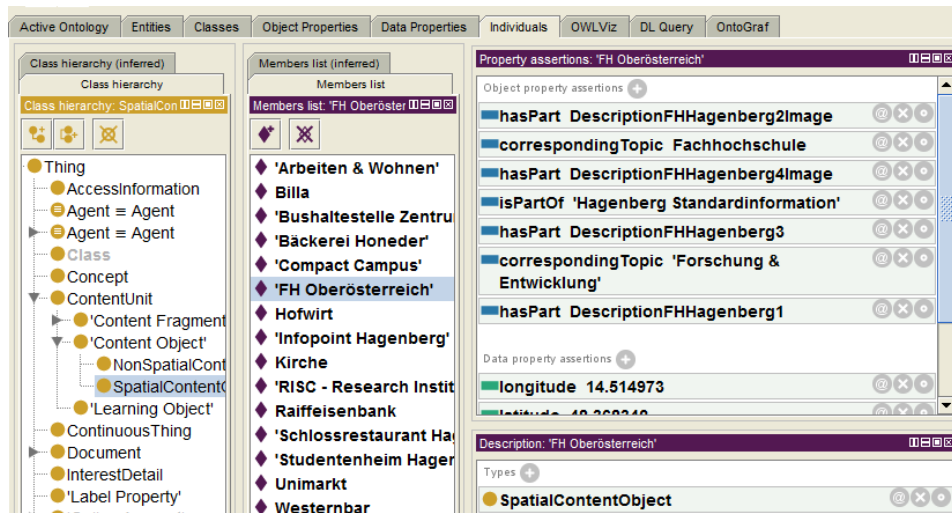


Abbildung 7.1: Erstellung eines Lernobjektes in Protege.

selbstgesteuerten Recherchen durch die betroffene Person. Letztere beziehen sich beispielsweise auf das Abrufen der Gemeindehomepage oder des Wikipedia-Eintrags über Hagenberg. Zusätzlich können Anwendungen wie Google-Maps miteinbezogen werden, die auch auf mobilen Geräten verfügbar sind und direkt im authentischen Kontext beispielsweise eine Umkreissuche nach gastgewerblichen Einrichtungen ermöglichen. Diese Informationsaufnahme findet unstrukturiert und mit Aufwand auf Lernerseite statt, weshalb hier durch das Mobile Learning Framework Verbesserungspotenzial und somit ein geeignetes Szenario besteht.

7.1.2 Befüllen der Wissensbasis

Der erste Schritt, um das System für dieses Szenario vorzubereiten, besteht darin, die Wissensbasis am Server mit Daten anzureichern, welche den in Abschnitt 5.3 vorgestellten Ontologien entsprechen. Dazu müssen in erster Linie Lernobjekte erstellt werden, welche Inhalte aufbereiten, die in weiterer Folge den Anwendern als Lerninhalte zur Verfügung gestellt werden. Zum Einpflegen dieser Inhalte kann beispielsweise der bereits vorgestellte Ontologie-Editor Protégé verwendet werden. Als Beispiel ist in Abbildung 7.1 ein in Protégé gestaltetes Lernobjekt abgebildet, das der Struktur der Lernobjekt-Ontologie entspricht.

Das linke Fenster zeigt einen Ausschnitt der Ontologie-Klassenhierarchie und gibt zusätzlich Aufschluss darüber, dass es sich beim hier vorgestellten Lernobjekt mit dem Namen „FH Oberösterreich“ um ein *SpatialContentObject* handelt. Dem rechten Fenster kann hingegen entnommen werden, dass das Lernobjekt aus insgesamt vier gereihten Teilen besteht, zwei Textkompo-

zenten und zwei Bildern. Zusätzlich ist es den Themengebieten „Forschung und Entwicklung“ sowie „Fachhochschule“ zugewiesen und Teil des übergeordneten Lerngebietes „Hagenberg Standardinformation“. Diese und weitere Informationen, wie die soeben genannten Themengebiete, können ohne Weiteres mit Protégé gestaltet und anschließend in den Triplestore importiert werden. Um die Lernobjekte zu strukturieren sollten sie zu übergeordneten Lerngebieten zusammengefasst und bei Bedarf mithilfe der Eigenschaften *precedes* und *follows* gereiht werden. Details zu den Möglichkeiten, welche von der Lernobjekt-Ontologie bereitgestellt werden, sind in Abschnitt 5.3.4 zu finden.

7.1.3 Client-Ablauf

Sobald die Wissensbasis des Systems befüllt ist, können Anwender sich mithilfe des in Abschnitt 6.2 erläuterten Android-Clients zum Server verbinden. Dazu muss in einem ersten Schritt der Benutzer für jeden Lernenden angelegt werden.

Registrierung und Anmeldung:

Die Registrierung eines neuen Benutzers kann entweder im Vorfeld parallel zum Anlegen der Lernobjekte geschehen oder durch die am Client vorhandene Registrierungs-Oberfläche, welche in Abbildung 7.2 dargestellt ist. Nach diesem einmaligen Registrierungsvorgang ist bei einem erneuten Ausführen der Anwendung nur noch eine normale Anmeldung mit Benutzernamen und Passwort notwendig.

Passive Auswahl der Lernobjekte:

Nachdem der Benutzer im System registriert ist, stellt der Client die normale Programmansicht dar und beginnt mit der Übermittlung von Kontext-Informationen, welche der Server anschließend in den Triplestore einpflegt. Basierend auf diesen Daten beginnt die Auswahl der Lernobjekte anhand des in Abschnitt 5.4 vorgestellten Schemas. Die Lernobjekte sind für dieses Szenario in einer besonderen Struktur angeordnet, welche unter anderem vorgibt, dass zuerst alle Lernobjekte des Lerngebietes „Hagenberg Einführung“ konsumiert werden müssen, bevor das System andere Lernobjekte zur Verfügung stellt. Innerhalb dieses Lerngebietes ist definiert, dass zuerst ein Lernobjekt mit einführenden Informationen über Hagenberg an den Client übermittelt werden muss.

Der Anwender wird über eine kleine Einblendung am Display in Begleitung mit einem Benachrichtigungston von den neu verfügbaren Informationen informiert, welche er anschließend in der Lernobjekt-Ansicht des Clients abrufen kann, die in Abbildung 7.3 dargestellt ist. Nach diesen einführenden

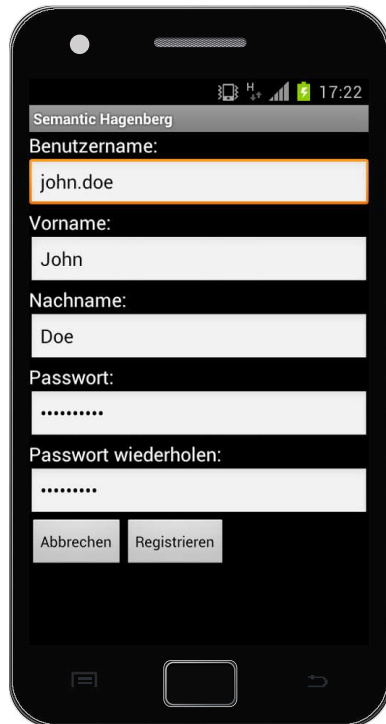


Abbildung 7.2: Oberfläche zur Registrierung.

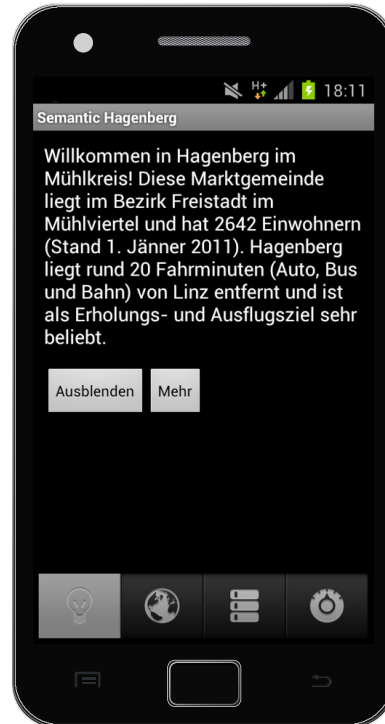


Abbildung 7.3: Einführendes Lernobjekt.

Informationen kann der Anwender auf weitere Lernobjekte warten (während er sich beispielsweise in der Umgebung Hagenberg bewegt), mithilfe der anderen Programmansichten Einstellungen anpassen oder gezielt nach weiterführenden Inhalten suchen.

Einstellungsmöglichkeiten

Das System stellt den Anwendern die Möglichkeit zur Verfügung, auf die serverseitige Auswahl der Lernobjekte direkt Einfluss zu nehmen ohne selbst Lernobjekte selektieren zu müssen. Dazu können in der Einstellungs-Ansicht, wie durch Abbildung 7.4 dargestellt, Interessen in vorgegebene Themengebiete adjustiert werden, wenn die automatische systemseitige Anpassung nicht als ausreichend erachtet wird. Beispielsweise sinkt durch eine geringere Gewichtung des Themengebiets „Politik“ die Wahrscheinlichkeit, dass Lernobjekte, welche dieser Kategorie zugewiesen sind, ausgewählt werden. Änderungen werden direkt in den Triplestore übertragen und in der weiteren Lernobjekt-Auswahl automatisch berücksichtigt. Die Kategorien werden bei der Erstellung der Wissensbasis definiert und sind speziell auf jene Lernobjekte, welche vom System verwaltet werden, zugeschnitten. Darüber hinaus

kann in dieser Ansicht die Maximalreichweite angepasst werden, innerhalb welcher ortsabhängige Lernobjekte Berücksichtigung finden. Natürlich nehmen die Lernenden auch indirekt über Änderungen der Kontextinformationen, beispielsweise durch Positionsänderungen oder die Erweiterung ihres Kenntnisstandes durch den Konsum von Lernobjekten, Einfluss auf die Auswahl der Lernobjekte.

Aktive Auswahl der Lernobjekte

Neben der serverseitigen Auswahl der Lernobjekte wird es den Lernenden auch ermöglicht, aktiv Lernobjekte auszuwählen. Dazu bietet die Kartenansicht eine gute Orientierungshilfe, welche den eigenen Standort und die Position von anderen Lernenden sowie von ortsabhängigen Lernobjekten erkennen lässt. Diese Übersichtskarte ist in Abbildung 7.5 dargestellt und zeigt zwei benachbarte Lernobjekte sowie einen weiteren Lernenden an, der sich in unmittelbarer Nähe befindet. Auf eines dieser Lernobjekte wurde bereits zugegriffen, weshalb es mit einem kleinen Haken anstatt des Fragezeichens versehen ist. Die Karteneinblendungen sind interaktiv und können vom Anwender dazu genutzt werden, Lernobjekte über einen Dialog von sich aus anzufordern und so den Lernvorgang zu einem gewissen Grad eigenständig zu steuern, wie in Abbildung 7.6 dargestellt ist. Ebenso kann auf diese Weise die grafische Einblendung anderer Benutzer verwendet werden, um ihnen Nachrichten zu übermitteln.

Eine weitere Möglichkeit, Lernobjekte anzufordern, wird durch die in Abbildung 7.7 dargestellte Listenansicht angeboten. Diese Auflistung enthält alle momentan vom System als für den Anwender relevant erachteten Lernobjekte. Die bereits aus der Einstellungs-Ansicht bekannten Kategorien dienen in dieser Bildschirmansicht zur Einteilung und gewähren eine bessere Übersichtlichkeit bei einer größeren Anzahl von Lernobjekten. Im Beispiel sind gerade ein Lernobjekt des Themengebietes „Fachhochschule“ und weitere, der Kategorie „Forschung & Entwicklung“ zugeordnete Lernobjekte verfügbar, wobei die letztgenannte Liste eingeklappt ist. Darüber hinaus ist es in dieser Ansicht möglich, bereits konsumierte Lernobjekte erneut anzufordern, indem mithilfe der Schaltfläche im oberen Bereich auf die Historie gewechselt wird.

7.2 Vergleich mit theoretischer Ausgangslage

Der entwickelte Lösungsansatz wird in diesem Abschnitt den in Kapitel 2 vorgestellten Grundlagen gegenübergestellt. Dabei wird evaluiert, ob Prinzipien eingehalten, Potenziale ausgenutzt und etwaige Einschränkungen ausreichend beachtet wurden.

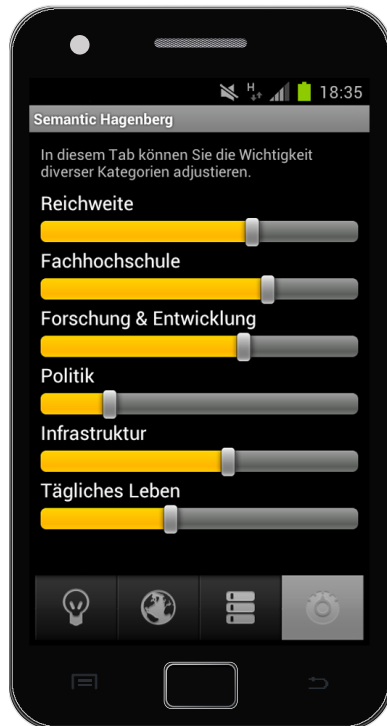


Abbildung 7.4: Einstellungsmöglichkeiten.

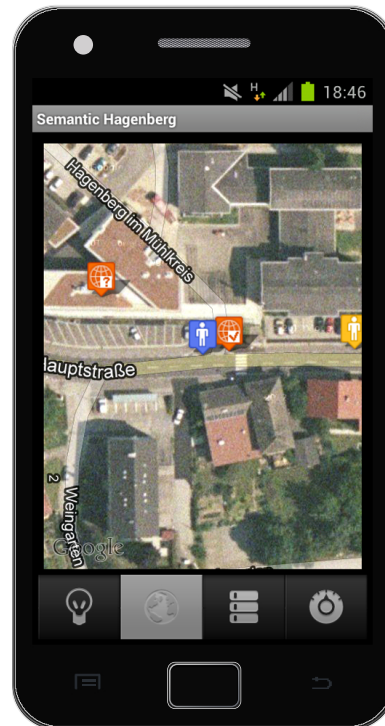


Abbildung 7.5: Kartenansicht mit Lernobjekten.

7.2.1 Einhaltung der Grundlagen von Mobile Learning

Grundsätzlich definiert sich Mobile Learning als Lernen, das an keinem vorgegebenen Standort unter Verwendung mobiler Technologien stattfindet. Insbesondere geht es um Prozesse, welche Informationen über die sich laufend verändernden Kontexte und Lernräume generieren und diese in weiterer Folge, vor allem beim Lernen in alltäglichen Situationen, sinnvoll nutzen [35], [22].

Dieser näheren Betrachtung von mobilem Lernen entspricht das vorgestellte Framework durchaus, weil das System speziell im Hinblick auf veränderte Informationen über den Kontext, in dem die Lernerfahrung stattfindet, entwickelt wurde und den Einsatz von Smartphones als mobile Technologie forciert. Die Verwaltung der Kontextinformationen in einem ontologiebasierten System bietet dabei Vorteile wie leichte Erweiterbarkeit, Schlussfolgerungs- und umfassende Abfragemechanismen. Ebenfalls kann das Framework gut für das Lernen in alltäglichen Situationen eingesetzt werden, wobei die Einschränkung besteht, dass ein definiertes Lernziel vorhanden sein muss, zu welchem Lernobjekte erstellt worden sind.

Eine mobile Lernanwendung sollte auch lerntheoretische Grundlagen be-



Abbildung 7.6: Anforderung eines Lernobjektes.

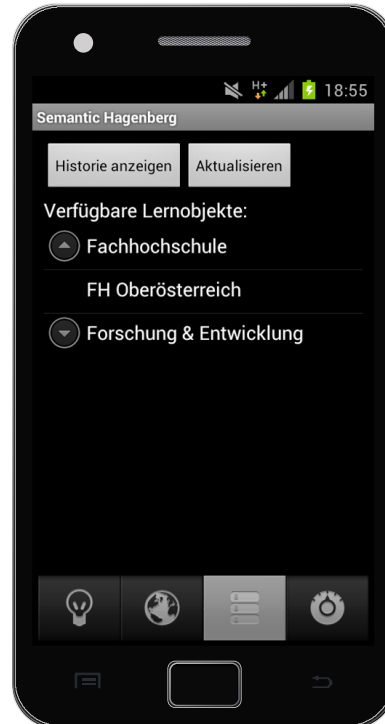


Abbildung 7.7: Liste aktuell verfügbarer Lernobjekte.

rücksichtigen, wie die von Ryu und Parsons [23] beschriebenen *three pillars of learning*, individuelles, situiertes und kollaboratives Lernen. Beim individuellen Lernen stehen die Lernenden im Mittelpunkt. Sie erarbeiten sich ihr Wissen selbst und weisen Eigenheiten auf, wie sie etwas lernen, wobei bestehendes Vorwissen und persönliche Erwartungen eine wichtige Rolle spielen. Das entwickelte System lässt den Lernenden möglichst viele Freiheiten, wodurch sie in der Lage sind, Lernobjekte eigenständig zu durchsuchen und auszuwählen. Darüber hinaus basiert auch die systemseitige Auswahl auf einer umfangreichen Anzahl von lernerspezifischen Kontextinformationen, wie beispielsweise dem Vorwissen in Form von bereits abgerufenen Lernobjekten oder vorhandenen Interessen.

Diese Kontextinformationen sind auch beim situierten Lernen von großer Bedeutung, welches definiert, dass der Lernprozess immer eingebettet in dem Kontext, in welchem er stattfindet, gesehen werden muss. Dabei ist es von Vorteil, wenn die Lernerfahrung in realistischen Situationen erfolgt, anstelle von dekontextualisierten Fortbildungsräumen. Der mobile Client ermöglicht das Lernen im authentischen Kontext und übermittelt unter anderem Positionsdaten an den Server, welcher so die Auswahl der Lernobjekte weiter

individualisieren kann.

Eine weitere Kategorie ist das kollaborative Lernen, welches besagt, dass Lernen in Gruppen Vorteile gegenüber anderen Methoden hat. Das vorgestellte Framework unterstützt dies nur insofern, dass Lernende die Möglichkeit haben, ihre Positionen gegenseitig in der Kartenansicht zu sehen und einander Nachrichten schicken zu können. Eine Möglichkeit, als Gruppe miteinander zu kommunizieren, wird nicht angeboten, was als Einschränkung des entwickelten Systems in seiner aktuellen Form gesehen werden kann.

7.2.2 Verwertung von Potenzialen

Mobile Learning bietet viele Potentiale, aus denen Nutzen gezogen werden kann. Diese werden in Abschnitt 2.3 näher erläutert, während in folgendem Teil näher darauf eingegangen wird, inwieweit sie vom entwickelten Framework verwertet werden.

Den technologischen Fortschritt und hohen Verbreitungsgrad betreffend werden bereits beim Client aktuelle Möglichkeiten ausgenutzt. Mit Android wird auf das derzeit am häufigsten verkaufte Smartphone-Betriebssystem [32] gesetzt, das eine Vielzahl von Sensoren unterstützt und für den Client verfügbar macht. Für andere Betriebssysteme existieren momentan keine Umsetzungen, die Schnittstellen am Server können jedoch auch von zukünftigen Entwicklungen betriebssystemunabhängig verwendet werden. Im Hinblick auf das Ausnutzen des technologischen Fortschritts ergeben sich neben clientseitigen Möglichkeiten, insbesondere durch die ontologiebasierte Informationsverwaltung am Server, Potentiale, wie etwa eine sehr kontextspezifische Informationsauswahl mit automatischen Schlussfolgerungen.

Mobile Lernanwendungen zeichnen sich vor allem dadurch aus, dass sie zeit- und ortsunabhängig eingesetzt werden können. Auch durch das vorgestellte System wird dieses Kriterium erfüllt und so ist es möglich, die Anwendung dort einzusetzen, wo die Lernerfahrung wirklich auftreten soll. Durch die Unterstützung von ortsabhängigen Lernobjekten ist ein Einsatz vor allem dann sinnvoll, wenn, wie im vorgestellten Szenario, über standortspezifische Inhalte gelernt werden kann. Die Ortsunabhängigkeit ist jedoch nicht ganz gegeben, weil der Client innerhalb von Gebäuden die Position nicht bestimmen kann. In diesem Bereich gibt es derzeit aber noch keine zufriedenstellenden Möglichkeiten ohne zusätzliche Hardware, wobei gegenwärtige Forschung auf eine Verbesserung hoffen lässt. Zeitliche Unabhängigkeit ermöglicht das Framework insofern, dass am Server abgelegte Informationen durchgehend gespeichert bleiben, auch wenn der Client zwischendurch nicht verwendet wird.

7.2.3 Beachtung der Einschränkungen

Doch wie in Abschnitt 2.4 beschrieben bietet mobiles Lernen im Vergleich zu konventionellem Lernen nicht nur Vorteile, es bringt auch eine Zahl von Einschränkungen mit sich, die berücksichtigt werden müssen.

Diesen Punkt betreffend sind vor allem technologische Barrieren zu erwähnen, wie kleinere Bildschirme und begrenzte Eingabemöglichkeiten. Erstere wirken sich in Kombination mit einem unruhigen Umfeld negativ auf die Konzentrationsfähigkeit aus, wobei die modular aufgebauten und zum Teil multimedialen Lernobjekte dieser Einschränkung zumindest teilweise entgegenwirken. Durch den modularen Aufbau wird die Menge an Informationen, die den Lernenden auf einmal präsentiert werden, begrenzt, wodurch der Lernvorgang weniger mentale Kapazitäten beansprucht. Trotzdem ergeben diese Bestandteile zusammen ein abgeschlossenes Inhaltselement, das eine größere Menge an Informationen beinhalten kann. Komplexere Inhalte können zusätzlich zum Text mit multimedialen Elementen wie Grafiken oder Audio angereichert werden, was sie für die Benutzer verständlicher gestaltet und diese weiter entlastet. Die begrenzten Eingabemöglichkeiten stellen weniger ein Problem dar, weil höchstens zur Kommunikation mit anderen Anwendern Eingaben erforderlich sind.

Demgegenüber besteht ein Defizit des Frameworks darin, dass die zu verwendenden Lernobjekte manuell erstellt werden müssen und dieser Vorgang vom System derzeit nicht unterstützt wird. Das stellt vor allem dann ein größeres Hindernis dar, wenn bereits vorhandene Ressourcen neu überarbeitet werden müssen.

7.3 Fazit

In Bezug auf die theoretischen Grundlagen des mobilen Lernens entspricht das vorgestellte System zu einem überwiegenden Teil den Vorgaben. Besonders die ontologiebasierte Datenhaltung zeichnet sich als optimale Grundlage für die Verwaltung der Kontextinformationen mit anschließender individualisierter Informationsauswahl aus, indem eine große Menge von Informationen über den Anwender und den Kontext, in welchem dieser sich befindet, strukturiert verwaltet und in komplexen Abfragen und Schlussfolgerungen genutzt werden können. Dabei werden Daten über den Lernenden und die Lernobjekte sowie deren Zeit-, Standort-, Aktivitäts- und Beziehungskontext berücksichtigt, wie vorhandenes Vorwissen, die Nähe zu ortsabhängigen Lernobjekten oder bestehende Interessen. Die entwickelten Ontologien sind darüber hinaus leicht erweiterbar, flexibel und wiederverwendbar, was sich positiv auf eventuelle zukünftige Erweiterungen des Systems auswirkt.

Ein weiterer Vorteil des Systems besteht vor allem darin, dass die Informationsauswahl vollkommen automatisch abläuft und die Lernenden sich so ganz auf jene Inhalte konzentrieren können, welche ihnen präsentiert wer-

den. Dennoch bestehen Möglichkeiten, um direkt Einfluss auf diesen Vorgang nehmen oder selbst Lernobjekte selektieren zu können, damit der Lernvorgang bei Bedarf auch selbstgesteuert durchgeführt werden kann. Auch die Möglichkeit, mit anderen Anwendern kommunizieren zu können, wird im Hinblick auf kollaboratives Lernen als positiv eingestuft.

Ein größeres Hindernis bei der Einführung des Systems wurde in der vorgestellten Fallstudie bei der Erstellung der Lernobjekte identifiziert. Für diesen Vorgang ist der Ontologie-Editor Protégé nur begrenzt geeignet, weil das Erstellen und Verknüpfen der Lernobjekte sowie deren einzelner Inhaltsfragmente sehr zeitaufwändig und fehleranfällig ist, weil leicht auf benötigte Eigenschaften vergessen werden kann. Eine Zwischenlösung würde die Entwicklung eines Editors darstellen, der das Initialisieren der Wissensbasis erleichtert, strukturiert und dabei verhindert, dass Inhalte unvollständig eingepflegt werden. Die aktuell bestmögliche Lösung wäre jedoch ein automatisches Befüllen des Triplestores mit Daten aus bestehenden Ressourcen, wie es beispielsweise mit *Ontology Learning*¹ umgesetzt werden kann.

Eine weitere Schwäche des Systems ist die sehr rudimentäre Umsetzung von Möglichkeiten des kollaborativen Lernens. Hier wäre es von Vorteil, wenn sich die Lernenden auch in Gruppen organisieren sowie austauschen könnten und generell eine Funktionalität eingeführt wird, mit der es einfacher ist, in Kontakt zu bleiben.

Zusammengefasst kann festgehalten werden, dass die vorrangigen Ziele erreicht wurden und insbesondere durch die ontologiebasierte Informationsverwaltung viele Systemfunktionen ermöglicht wurden, die beispielsweise bei der Verwendung einer rationalen Datenbank nicht möglich oder unverhältnismäßig komplex gewesen wären. Dennoch wurden einzelne Schwachstellen identifiziert, wie die unzureichende Unterstützung des kollaborativen Lernens und der zu aufwändige Erstellungsprozess von Lernobjekten, denen in eventuellen Weiterentwicklungen entgegengewirkt werden sollte.

¹Ein erster Einblick kann beispielsweise unter http://en.wikipedia.org/wiki/Ontology_learning gefunden werden.

Kapitel 8

Schlussbemerkungen

Vor allem Veränderungen in der Wirtschaft führen zu steigenden Anforderungen an Wissen, Kompetenzen, Flexibilität und Mobilität der Menschen. Fort- und Weiterbildungsmaßnahmen gewinnen zunehmend an Bedeutung und machen es sinnvoll, sich mit Ergänzungen zu konventionellen Lernmethoden zu beschäftigen, um den Herausforderungen lebenslangen Lernens gerecht werden zu können. Im Hinblick darauf stellt mobiles Lernen einen vielversprechenden Trend dar, der vor allem durch die hohe Verbreitung und den umfangreichen Funktionsumfang mobiler Endgeräte wie Smartphones möglich ist. Ebenso führen Forschungsarbeiten im Bereich des semantischen Webs zu verbesserten Möglichkeiten Wissen abzubilden, weshalb eine Kombination dieser Ansätze vielversprechend erscheint.

8.1 Zusammenfassung

Im Rahmen dieser Arbeit wurde ein Mobile Learning Framework entwickelt, das auf aktuellen Erkenntnissen im Bereich des mobilen Lernens basiert. Insbesondere dienen Informationen über den Kontext, in welchem die Lernerfahrung stattfindet, zur Individualisierung des Lernvorganges. Diese Kontextinformationen werden in einer serverseitigen, ontologiebasierten Datenbank verwaltet, wodurch spezielle Strukturierungs-, Schlussfolgerungs- und Abfragemechanismen genutzt werden können, um Lernobjekte automatisiert zur Verfügung zu stellen. Zur Erzeugung der Kontextinformationen werden einerseits Sensoren des mobilen Endgeräts verwendet, andererseits Rückschlüsse aus vorhandenen Daten gezogen. Insbesondere die Berücksichtigung des ortsbezogenen Kontextes und dessen Gewichtung grenzt diese Arbeit von verwandten Ansätzen ab.

Die ontologiebasierte Datenhaltung basiert auf vier Teilontologien, welche Informationen über den Lernenden, die Lernobjekte, den zeitlichen und den örtlichen Kontext verwalten. In Kombination sind die darin enthaltenen Daten ausreichend, um abhängig von der aktuellen Situation, in welcher sich

die Lernenden befinden, systemseitig reagieren zu können. Anhand der bekannten Kontextinformationen werden relevante Lernobjekte vorausgewählt und anschließend gereiht, um das für die jeweiligen Gegebenheiten am besten passende Lernobjekt zu ermitteln. Die Vorauswahl ist relativ komplex und basiert auf logischen Schlussfolgerungen sowie SPARQL-Abfragen, während die Reihung vor allem die Interessen der Lernenden einbezieht und einfacher gestaltet ist.

Das erstellte Framework umfasst einen Server, welcher die ontologiebasierte Datenhaltung übernimmt und Zugriffe auf verwaltete Informationen durch Schnittstellen ermöglicht, während der Client zur Generierung der Kontextinformationen eingesetzt wird und natürlich die Lernenden dazu befähigt, Funktionalitäten des Systems verwenden zu können. Trotz der gewünschten Automatisierung wurde bei der Gestaltung des Clients darauf geachtet, den Lernenden Möglichkeiten zur Verfügung zu stellen, um auf den Lernvorgang Einfluss zu nehmen oder diesen überwiegend selbstständig steuern zu können.

8.2 Resümee

Mit dem in dieser Arbeit vorgestellten System wurden die gesetzten Ziele erreicht und dabei die Grundlagen mobilen Lernens eingehalten. Zur Umsetzung dieser Rahmenbedingungen hat sich der Einsatz einer ontologiebasierten Wissensbasis angeboten und im Verlauf der Entwicklung sowie Evaluierung als günstiger Ansatz herausgestellt. Diese Lösung ermöglicht eine strukturierte, erweiterbare und wiederverwendbare Verwaltung der Kontextinformationen, die mithilfe von Werkzeugen des semantischen Webs wie OWL und SPARQL auch für Schlussfolgerungen eingesetzt werden kann. Auf diesen Technologien basiert auch die automatisierte Auswahl von Inhalten für die Lernenden.

Das Framework kann in einer Vielzahl von Anwendungsgebieten eingesetzt werden, wobei es sein Potential vor allem dann entfalten kann, wenn eine gewisse Anzahl von ortsbezogenen Lerninhalten existieren. Dies ist beispielsweise beim gezielten Kennenlernen von Betriebsanlagen oder touristischen Zielen gegeben. Nach dem Einrichten des Servers müssen nur noch die Lernobjekte erstellt werden, woraufhin sich die Lernenden nach der Installation des Clients beim System anmelden und mit dem Lernvorgang beginnen können.

In seiner aktuellen Form liegen die Schwachstellen des Frameworks vor allem bei der Erstellung von Lernobjekten. Dieser Prozess ist derzeit sehr zeitaufwändig und anfällig für Fehler, weshalb alternative Möglichkeiten miteinbezogen werden sollten. Darüber hinaus wäre eine verbesserte Unterstützung des kollaborativen Lernens von Vorteil. Grundsätzlich überwiegen jedoch die positiven Aspekte und der vorgestellte Ansatz stellt ein lauffähiges

und praktisch einsetzbares System dar, das zusätzlich als Ausgangslage für Weiterentwicklungen, welche den identifizierten Schwachstellen entgegenwirken, gesehen werden kann.

8.3 Ausblick

Die steigende Verbreitung von Smartphones, bessere technische Möglichkeiten und der zunehmende Forschungsstand, vor allem in betrieblichen und akademischen Bereichen, lassen darauf schließen, dass Mobile Learning als Ergänzung zu konventionellen Lern- und Lehrmethoden in weiterer Zukunft an Bedeutung gewinnt. Ebenfalls schreiten Entwicklungen im Bereich des semantischen Webs voran, wobei Standards weiterentwickelt und verbesserte Tools sowie Frameworks verfügbar werden. Mittlerweile erkennen und nützen bereits Unternehmen wie Google die Vorteile von semantischen Ansätzen¹, was gleichermaßen auf positive Entwicklungen schließen lässt.

Durch die Kombination der beiden Ansätze ist ein vielversprechendes Framework entstanden, welches zeigt, dass die Integration dieser Technologien durchaus Potential bietet. Insbesondere zur Gestaltung der Kontextinformationen und für die darauf basierende Auswahl von Lernobjekten ist der Einsatz einer ontologiebasierten Informationsverwaltung von Vorteil, was auch für zukünftige Entwicklungen gilt.

Wie die Evaluierung zeigt, sollten in eventuellen Weiterentwicklungen des vorgestellten Systems Möglichkeiten angeboten werden, um eine bessere Kommunikation zwischen den Lernenden zu ermöglichen und das Einpflegen von Lernobjekten zu vereinfachen. Letzteres ist in naher Zukunft vermutlich zunehmend automatisiert möglich, wenn beispielsweise Methoden des Ontology Learning weiter fortgeschritten sind. Die aktuelle Umsetzung bietet darüber hinaus zusätzliche Erweiterungsmöglichkeiten, wie die Anzeige des aktuellen Lernfortschritts oder das Einbeziehen zusätzlicher Kontextinformationen.

¹Google setzt bereits semantische Suche ein und ermöglicht dadurch beispielsweise das Unterscheiden zwischen gleichen Begriffen mit unterschiedlichen Bedeutungen. Siehe <http://online.wsj.com/article/SB10001424052702304459804577281842851136290.html> für weitere Informationen.

Anhang A

Inhalt der CD-ROM

Format: CD-ROM, Single Layer, ISO9660-Format

A.1 PDF-Dateien

Pfad: /

Pichler_Alexander_2012.pdf Digitale Version vorliegender Arbeit

A.2 Online-Quellen

Pfad: /Online-Quellen/

32Gartner.pdf Worldwide Sales of Mobile Phones Declined 2
Percent in First Quarter of 2012

33BergInsight.pdf Handset Connectivity Technologies

34JelenaJovanovic.pdf ALOCoM Core Ontology

35OMalley.pdf Guidelines for Learning/Teaching/Tutoring
in a Mobile Environment

36EricPrudhommeaux.pdf SPARQL Query Language for RDF

37HannesWeger.pdf Mehr Handys als Menschen:
Durchdringungsrate in Österreich bereits
über 100 Prozent

A.3 Projektdateien

Pfad: /project/

source/ Quellcode

executables/ Ausführbare Dateien

Dokumentation.pdf Dokumentation und Installationsanleitung

Quellenverzeichnis

Literatur

- [1] Damiao Ribeiro de Almeida, Claudio de Souza Baptista und Fabio Gomes de Andrade. „Using Ontologies in Context-Aware Applications“. In: *Proceedings of the 17th International Conference on Database and Expert Systems Applications (DEXA '06)*. Hrsg. von Stéphane Bressan, Josef Küng und Roland Wagner. 2006, S. 349–353.
- [2] Anne K. Bednar u. a. „Theory into Practice: How Do We Link?“. In: *Constructivism and the Technology of Instruction: A Conversation*. Hrsg. von Thomas M. Duffy und David H. Jonassen. Hillsdale: Lawrence Erlbaum Associates, 1992, S. 17–34.
- [3] Steve Benford. *Future Location-Based Experiences*. Techn. Ber. 1. JISC Technology und Standards Watch, 2005. URL: http://www.jisc.ac.uk/media/documents/techwatch/jisctsw_05_01.pdf.
- [4] Jawad Berri, Rachid Benlamri und Yacine Atif. „Ontology-based Framework for Context-aware Mobile Learning“. In: *Proceedings of the 2006 International Conference on Wireless Communications and Mobile Computing (IWCMC '06)*. Hrsg. von Seizo Onoe u. a. 2006, S. 1307–1310.
- [5] Barry Bishop u. a. „OWLIM: A family of scalable semantic repositories“. In: *Semantic Web 2* (2011), S. 33–42.
- [6] Yilun Chia u. a. „Context-Aware Mobile Learning with a Semantic Service-Oriented Infrastructure“. In: *Proceedings of the 2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications (WAINA '11)*. 2011, S. 896–901.
- [7] Anind K. Day. „Understanding and Using Context“. In: *Personal and Ubiquitous Computing Journal* 5 (2001), S. 4–7.
- [8] Günther Dohmen. *Das informelle Lernen: Die internationale Erschließung einer bisher vernachlässigten Grundform menschlichen Lernens für das lebenslange Lernen aller*. Bonn: Bundesministerium für Bildung und Forschung (BMBF), 2001. URL: <http://www.werkstatt->

- frankfurt.de/fileadmin/Frankfurter_Weg/Fachtagung/BMBF_Das_informelle_Lernen.pdf.
- [9] Dirk Frohberg. „Mobile Learning“. Diss. Zürich: Wirtschaftswissenschaftliche Fakultät der Universität Zürich, 2008.
- [10] Pascal Hitzler u. a. *Semantic Web: Grundlagen*. Berlin: Springer, 2008.
- [11] Ton de Jong. „Knowledge Construction and Sharing with Media Based Applications“. In: *Campus 2002: Die Virtuelle Hochschule in der Konsolidierungsphase*. Hrsg. von Gudrun Bachmann, Odette Haefeli und Michael Kindt. Münster: Waxmann, 2002, S. 431–442.
- [12] Desmond Keegan. „Mobile Learning: The Future is Wireless“. In: *Handbuch E-Learning*. Hrsg. von Andreas Hohenstein und Karl Wilbers. 9. Ergänzungslieferung Juli 2004. Köln: Deutscher Wirtschaftsdienst, 2010, S. 1252–1253.
- [13] Werner Kohn. „E-Learning bei der D.A.S. Versicherung“. In: *Handbuch E-Learning*. Hrsg. von Andreas Hohenstein und Karl Wilbers. Grundwerk Dezember 2001. Köln: Deutscher Wirtschaftsdienst, 2010, S. 2535–2556.
- [14] Peter Kraus-Hoffmann und Maciej A. Kuszpa. *Mobile Learning: Grundlagen und Perspektiven*. Techn. Ber. 24. Initiative Neue Qualität der Arbeit (INQA), 2007. URL: http://www.inqa.de/SharedDocs/PDFs/DE/Publikationen/inqa-24-mobile-learning.pdf?__blob=publicationFile.
- [15] Maciej A. Kuszpa. „Manager lernen mit dem Handy - Erfahrungen mit mobilem E-Learning in der Weiterbildung von Führungskräften“. In: *Handbuch E-Learning*. Hrsg. von Andreas Hohenstein und Karl Wilbers. 13. Ergänzungslieferung August 2005. Köln: Deutscher Wirtschaftsdienst, 2010, S. 1859–1861.
- [16] Jean Lave und Etienne Wenger. *Situated Learning: Legitimate Peripheral Participation*. Cambridge: Cambridge University Press, 1991.
- [17] John McCarthy. „Notes on formalizing contexts“. In: *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI '93)*. Hrsg. von Ruzena Bajcsy. 1993, S. 555–560.
- [18] Moeiz Miraoui, Chakib Tadj und Chokri ben Amar. „Context Modeling and Context-Aware Service Adaptation for Pervasive Computing Systems“. In: *International Journal of Computer and Information Science and Engineering* 2 (2008), S. 567–576.
- [19] Philip Moore u. a. „A Survey of Context Modelling for Pervasive Cooperative Computing“. In: *IEEE Proceedings of the First International Symposium on Information Technologies and Applications in Education (ISITAE '07)*. Hrsg. von Yuan Kunming und P. R. China. 2007, S. K5/1–K5/6.

- [20] Laura Naismith u. a. *Literature Review in Mobile Technologies and Learning*. Techn. Ber. 11. Futurelab, 2004. URL: http://www.futurelab.org.uk/sites/default/files/Mobile_Technologies_and_Learning_review.pdf.
- [21] Reinhard Oppermann und Marcus Specht. „Lernen im Betrieb mit M-Learning gestalten“. In: *Handbuch E-Learning*. Hrsg. von Andreas Hohenstein und Karl Wilbers. 5. Ergänzungslieferung August 2003. Köln: Deutscher Wirtschaftsdienst, 2010, S. 1228–1248.
- [22] Norbert Pachler, Ben Bachmair und John Cook. *Mobile Learning: Structures, Agency, Practices*. London: Springer, 2010.
- [23] Hokyoung Ryu und David Parsons. „Designing Learning Activities with Mobile Technologies“. In: *Innovative Mobile Learning: Techniques and Technologies*. Hrsg. von Hokyoung Ryu und David Parsons. London: Information Science Reference, 2009, S. 1–20.
- [24] Toby Segaran, Colin Evans und Jamie Taylor. *Programming the Semantic Web*. Sebastopol: O’Reilly, 2009.
- [25] Mike Sharples, Josie Taylor und Giasemi Vavoula. „A Theory of Learning for the Mobile Age“. In: *The Sage Handbook of E-Learning Research*. Hrsg. von Richard Andrews und Carolie Haythornthwaite. London: Sage Publications, 2007, S. 221–247.
- [26] Melody Siadaty u. a. „m-LOCO: An Ontology-based Framework for Context-Aware Mobile Learning“. In: *Proceedings of the 6th International Workshop on Ontologies and Semantic Web for E-Learning (SWEL ’08) at 9th International Conference on Intelligent Tutoring Systems (ITS ’08)*. 2008, S. 40–49.
- [27] Yanjie Song. „Handheld Educational Applications: A Review of the Research“. In: *Innovative Mobile Learning: Techniques and Technologies*. Hrsg. von Hokyoung Ryu und David Parsons. London: Information Science Reference, 2009, S. 302–323.
- [28] Thomas Strang und Claudia Linnhoff-Popien. „A Context Modeling Survey“. In: *Workshop on Advanced Context Modelling, Reasoning and Management, The Sixth International Conference on Ubiquitous Computing (UbiComp ’04)*. 2004, S. 31–41.
- [29] Rudi Studer, V. Richard Benjamins und Dieter Fensel. „Knowledge Engineering: Principles and Methods“. In: *Data and Knowledge Engineering* 25 (1998), S. 161–197.
- [30] John Traxler. „Current State of Mobile Learning“. In: *Mobile Learning: Transforming the Delivery of Education & Training*. Hrsg. von Mohamed Ally. Edmonton: AU Press, 2009, S. 9–24.

- [31] Andreas Zimmermann, Andreas Lorenz und Reinhard Oppermann. „An Operational Definition of Context“. In: *Proceedings of the 6th International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT '07)*. Hrsg. von Boicho N. Kokinov u. a. 2007, S. 558–571.

Online-Quellen

- [32] *Gartner Says Worldwide Sales of Mobile Phones Declined 2 Percent in First Quarter of 2012; Previous Year-over-Year Decline Occurred in Second Quarter of 2009*. Kopie auf CD-Rom (32Gartner.pdf). Gartner. 2012. URL: <http://www.gartner.com/it/page.jsp?id=2017015>.
- [33] *Handset Connectivity Technologies*. Kopie auf CD-Rom (33BergInsight.pdf). Berg Insight. 2012. URL: <http://www.berginsight.com/ReportPDF/ProductSheet/bi-hct3-ps.pdf>.
- [34] Jelena Jovanovic u. a. *ALOCOM Core Ontology*. Kopie auf CD-Rom (34JelenaJovanovic.pdf). 2010. URL: <http://jelenajovanovic.net/ontologies/loco/alocom-core/spec/>.
- [35] Claire O'Malley u. a. *Guidelines for Learning/Teaching/Tutoring in a Mobile Environment*. Kopie auf CD-Rom (35OMalley.pdf). Mobilearn. 2003. URL: <http://www.mobilearn.org/download/results/guidelines.pdf>.
- [36] Eric Prud'hommeaux und Andy Seaborne. *SPARQL Query Language for RDF*. Kopie auf CD-Rom (36EricPrudhommeaux.pdf). W3C (World Wide Web Consortium). 2008. URL: <http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/>.
- [37] Hannes Weger. *Mehr Handys als Menschen: Durchdringungsrate in Österreich bereits über 100 Prozent*. Kopie auf CD-Rom (37HannesWeger.pdf). 2006. URL: <http://www.presetext.at/news/060719029/mehr-handys-als-menschen>.