

Entwicklung von gekoppelten Lernumgebungen mit Interoperabilitätsstandards

BACHELORARBEIT

KARLSRUHER INSTITUT FÜR TECHNOLOGIE

Kansei Hara

15. September 2019

Verantwortlicher Betreuer: Prof. Dr.-Ing. J. Beyerer
Prof. Dr. rer. nat. H. Steusloff

Betreuender Mitarbeiter: Dipl.-Inf. Alexander Streicher

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Bachelorarbeit ohne unzulässige fremde Hilfe selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt zu haben.

Karlsruhe, den 15. September 2019

(Kansei Hara)

Inhaltsverzeichnis

1	Abstract	1
2	Kurzfassung	3
3	Einleitung	5
3.1	Motivation	5
3.2	Zielsetzung	6
3.3	Aufbau	6
4	Grundlagen	7
4.1	Lernmanagementsysteme	7
4.2	Das Serious Game „Lost Earth“	7
4.3	Interoperabilitätsstandards im Bereich E-Learning	8
4.3.1	Überblick über die Interoperabilitätsstandards	9
4.3.2	Learning Tools Interoperability	11
4.3.3	xAPI	15
4.4	Unity	16
5	Stand der Forschung und Technik	19
5.1	Kopplung von Lernmanagementsystemen	19
5.2	Kopplung von Lernplattformen mit Lernspielen	19
5.3	Unity und xAPI	20
6	Konzeption	21
6.1	Anforderungsanalyse	21
6.1.1	Überblick	21
6.1.2	Anforderungen	22
6.2	Bestimmen der Komponenten und Interoperabilitätsstandards	23
6.2.1	Komponenten	24
6.2.2	Interoperabilitätsstandards	25

6.3	Kompletter Anwendungsablauf	26
6.4	Architektur	31
7	Implementierung	33
7.1	Priorisierung der Anforderungen	33
7.2	Umsetzung der Anforderungen	34
7.2.1	Proof-of-Concept	34
7.2.2	Umsetzung 1: Anzeigen der externen Lerninhalte	35
7.2.3	Umsetzung 2: Integration des LTI-Lern-Tools	37
7.2.4	Umsetzung 3: Integration des LRS	40
7.2.5	Umsetzung 4: Übertragung der Benutzerdaten an das LE	41
7.2.6	5. Umsetzung: Anbieten der externen Lernmaterialien für das LE	43
7.2.7	Umsetzung 5: Einfluss der Ergebnisse in Lost Earth	44
7.3	Umsetzung in LE	44
8	Anwendungsszenario	49
9	Fazit und Ausblick	57
9.1	Ergebnis	57
9.2	Diskussion und Ausblick	57
9.3	Fazit	58
	Literatur	59
	Abbildungsverzeichnis	65

1 Abstract

Nowadays, different learning environments can communicate with each other and form a large learning environment. In learning network conventional Serious Games are often only the end of a coupling and have little added value compared to learning platforms. The aim of this work is to develop a technical implementation of a learning environment based on a serious game that is linked to existing and independent learning environments, taking into account the current and established interoperability standards. After implementation, a learning platform will provide an external learning game. At the same time, the game acts as a learning platform and offers further external tools. It can also store user activities and read activities created by other learning environments. The Serious Game thus extends the learning environment even further.

2 Kurzfassung

Ziel dieser Arbeit ist es unter Einbeziehung der aktuellen und etablierten Interoperabilitätsstandards, eine Konzeption und eine technische Realisierung einer Kopplung von mehreren existierenden und unabhängigen Lernumgebungen zu entwickeln. Mit der resultierenden Vereinigung der Lernumgebungen kann man die Vorteile der verschiedenen Umgebungen kombinieren und erzielt zusätzlich einen reibungslosen Lernprozess, ohne dass man beispielsweise sich für jede Umgebung wiederholt anmelden muss. Für die Umsetzung wird die vom Fraunhofer angebotene Plattform mit dem Serious Game Lost Earth, das ebenfalls vom Fraunhofer entwickelt wurde, gekoppelt. Dieses Spiel wird wiederum mit ein externes Lern-Tool und einen Learning Record Store(LRS) gekoppelt. Das Lern-Tool bietet Lernmaterialien oder Inhalte für externe Lernumgebungen zur Verfügung. Der LRS ist dafür da um Benutzeraktivitäten, beispielsweise abgeschlossene Kurse oder Ergebnisse, mit anderen Lernumgebungen zu teilen und wird mit xAPI-Standard dargestellt. Die Umsetzung für das Aufrufen von externe Tools wird der LTI-Standard verwendet und für das Tracken der Benutzeraktivitäten der xAPI-Standard. Diese Standards werden ausführlich in der Arbeit untersucht. Anschließend einen Grundkonzept für die Umsetzung ausgearbeitet. Dazu werden die Anforderungen aufgestellt, den Ablauf eines Lernenden im System dargestellt und die einzelnen Komponente näher erläutert. Zum Schluss wird die Implementierung dokumentiert und das Ergebnis evaluiert. Nach der Realisierung stellt die Lernplattform das externe Lernspiel Lost Earth bereit. Das erweiterte Lernspiel agiert nun wie eine Lernplattform und bietet weitere austauschbare externe Lern-Tools an. Zusätzlich kann das Spiel Benutzeraktivitäten vom LRS abrufen und eigene Aktivitäten speichern. Damit stellt es die Möglichkeit bereiten mit weiteren Lernumgebungen oder Tools zu kommunizieren.

3 Einleitung

Heutzutage werden Lehr- und Lernprozesse mit digitalen Medien oder Werkzeugen unterstützt und sind nicht mehr wegzudenken. Diese elektronische Unterstützung wird als E-Learning bezeichnet. Dadurch können laut [TV97] die Lehrenden kostengünstig ihr Wissen digital publizieren und die Lernenden sich dieses Wissen flexibler aneignen und ihre Fähigkeiten verbessern.

Der englische Begriff 'E-Learning' beschreibt alle elektronischen Formen des Lernens. Die Angebote sind heutzutage enorm und ein großer Teil davon sind beispielsweise Lernplattformen. Auf diese Lernumgebungen werden von verschiedenen Mitwirkenden Lerninhalte und Kurse publiziert und für Lernende angeboten. Beispielsweise bietet Moodle.net für über 150 Millionen Benutzer fast 20.000 Kurse an [Moo19]. Weitere große Lernplattformen sind Open Edx oder Blackboard. Ferner sind diese Plattformen trotz ihrer großen Skalierbarkeit flexibel und vielfältig [Row94]. Ein weiterer, aber kleiner Teil des E-Learnings sind Serious Games [Pop+11]. Diese sind Videospiele mit der Hauptaufgabe, einen Lerneffekt zu erzielen, beispielsweise in Mathematik, Physik, Ingenieurwesen, Medizin, Wissenschaft, Geschichte oder Literatur [Squ+04; Eva+15; MJ09; IFF13]. Die Besonderheit von Lernspielen ist, dass die Aufnahmekraft des Lernstoffs und ihre motivierende Anziehungskraft stärker sind als das herkömmliche Empfangen von Wissen. Ein Grund dafür ist die interaktive Beschäftigung mit den Lerninhalten (Learning by doing) [FMK11].

3.1 Motivation

Durch die Vielfalt des E-Learnings ist es oft notwendig, mit mehreren heterogenen Systemen zu kommunizieren. Ohne das Festlegen eines Standards für die Inhalte und E-Learning-Strukturen ist es nicht möglich, diese Interoperabilität zu gewährleisten [Bak+17]. Mit einem E-Learning-Interoperabilitätsstandard und den dazugehörigen Spezifikationen ist es möglich, einheitliche Protokolle oder Dateiformate zu setzen, die die Zusammenarbeit zwischen Systemen bewältigen trotz unterschiedlichen Sprachen, Schnittstellen und Plattformen [Weg96].

3.2 Zielsetzung

Ziel dieser Arbeit ist es unter Einbeziehung der aktuellen und etablierten Interoperabilitätsstandards, eine Konzeption und eine technische Realisierung einer Kopplung von mehreren existierenden und unabhängigen Lernumgebungen zu entwickeln. Mit der resultierenden Vereinigung der Lernumgebungen kann man die Vorteile der verschiedenen Umgebungen kombinieren und erzielt zusätzlich einen reibungslosen Lernprozess, ohne dass man beispielsweise sich für jede Umgebung wiederholt anmelden muss.

Konkreter bietet eine webbasierte Lernplattform ein Serious Game an. Vorteil bei dieser Kopplung ist es, dass das Lernspiel systemunabhängig und ohne das gesamte Spiel herunterzuladen, angeboten wird. Durch das Integrieren einer spielerischen Komponente auf der Lernplattform, wird die Motivation des Lernenden gestärkt und hat somit eine positive Auswirkung auf das Lernen. Zusätzlich werden Lerninhalte im Spiel mit weiteren Lernmaterial von einer oder mehreren Lernplattformen angeboten, um die Wartbarkeit und Quantität des Spielinhaltes zu verbessern und zu erhöhen. Für die Datenkommunikation des Lernfortschrittes zwischen den unterschiedlichen Systemen werden die Fortschritte zuerst erfasst und auf einer externen, speziell dafür entwickelten Datenbank gespeichert und abrufbar gemacht.

3.3 Aufbau

Als erstes werden im folgenden Kapitel die Hintergründe, worauf diese Arbeit aufbaut, erläutert und auf die Grundlagen eingegangen, wie eine allgemeine Kopplung der verschiedenen Systeme zu realisieren sind. Anschließend werden derzeitige gekoppelte Lernumgebungen untersucht und analysiert. In nächsten Kapitel wird die Konzeption der Kopplung vorgestellt. Hierbei werden die Anforderungen definiert, ein Grundkonzept ausgearbeitet und die Gesamtarchitektur erstellt. Auf die Gesamtarchitektur angelehnt, werden die Anforderungen nach und nach mit SCRUM implementiert. Zum Schluss wird ein Fazit gezogen und ein Ausblick vorgeschlagen.

4 Grundlagen

In diesem Kapitel wird ein Überblick über die aktuellen und bereits etablierten Spezifikationen und Standards der Interoperabilität dargestellt [Bak+17] und Grundlegendes der einzelnen Lernumgebungen des E-Learnings beschrieben.

4.1 Lernmanagementsysteme

Ein Lernmanagementsystem ist ein Webservice für das E-Learning. Einer der ersten Grundbausteine wurde in den späten 1990er Jahre gelegt [Bet09]. In diesem System werden elektronisch Lernangebote als Einzel- oder Sammelwebservice angeboten. Auf diesen Systemen können Lerninhaltsanbieter für Lernende Lernmaterialien erstellen, speichern, organisieren, bearbeiten und publizieren [VW03]. Durch die Digitalisierung der Lernmaterialien und die dadurch resultierende höhere Reichweite fördern solche Lernumgebungen das lebenslange und flexible Lernen [Sim18].

Heute, werden unzählige Lernmanagementsysteme bereitgestellt und die Reichweite ist enorm. Beispielsweise bietet Moodle.net für über 150 Millionen Benutzer fast 20.000 Kurse an [Moo19]. Die besten Plattformen 2018, darunter auch Moodle.net, werden in der Tabelle 4.1 aufgelistet [Lou].

4.2 Das Serious Game „Lost Earth“

Serious Games sind Lernspiele, mit der Hauptaufgabe, einen positiven Lerneffekt zu erzielen [DA]11]. Diese Spiele basieren laut einem umfassenden Überblick über Lernspiele von Kebritchi und Hirumi auf verschiedenen Lerntheorien wie „Learning by Doing“, „Discovery Learning“ oder „Constructivism Learning“ [KH08]. In Bereichen wie Militär, Gesundheit etc. werden diese erfolgreich angewandt, allerdings ist der Einsatz im staatlichen Bildungssystem, also von der Grundschule bis zur Universität, gering [Pop+11]. Lost Earth ist ein vom Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung entwickeltes Serious Game mit dem Ziel, die Bildauswertung anhand von realem Bildmaterial einzuüben. Zusätzlich werden

Top 5 der cloud-basierten Lernmanagementsysteme	
1.	Docebo
2.	Adobe Captivate Prime
3.	Litmos LMS
4.	iSpring Learn
5.	Talent LMS

Top 5 der Open-Source-Lernmanagementsysteme	
1.	Moodle
2.	Chamilo
3.	Totara Learn
4.	Canvas
5.	Open edX

Tabelle 4.1: Rangliste der besten Lernplattformen [Lou]

Aufklärungsabläufe vertieft und der Umgang damit durch Sensoren und Sensorplattformen verbessert.

4.3 Interoperabilitätsstandards im Bereich E-Learning

Unter dem Begriff Interoperabilitätsstandard versteht man eine Norm, wodurch mehrere unabhängige Systeme trotz unterschiedlicher Sprachen, Schnittstellen und Plattformen miteinander zusammenarbeiten können [Weg96].

Im Jahr 2002 gab es mehrere Standards im Anwendungsbereich des E-Learnings. Collier und Robson forderten in den darauffolgenden Jahren auf, dass die E-Learning-Standards, die gut definiert waren und oft Verwendung fanden, sich durchsetzen mussten, um die Vorteile und somit auch die Ziele der Interoperabilitätsstandards, weitläufig nutzen zu können [CRC02]. In der Abbildung 4.1 sind die derzeit durchgesetzten Standards abgebildet [Bak+17].

Ziel der Standards im E-Learning-Bereich sind:

Statt des Entwickelns verschiedener Schnittstellen für unterschiedliche Systeme, wird eine allgemeingültige Schnittstelle von der standardisierten Interoperabilität zur Verfügung gestellt. Dies führt zur geringeren Implementationskosten.

Die Produzenten müssen nur einmalig Inhalte für einen bestimmten Standard erstellen, damit diese von allen Systemen verwendet werden können.

Für die Lernenden gibt es mehr Auswahl an Produkten und diese weisen eine höhere Portabilität auf, wodurch das Lernen unterstützt wird.

4.3.1 Überblick über die Interoperabilitätsstandards

In einer E-Learning-Anwendungsumgebung gibt es verschiedene Anforderungen und somit verschiedene Objekte und Informationen, die untereinander ausgetauscht werden [WG09]. Daraus resultieren unterschiedliche Standardtypen für die jeweiligen Anwendungszwecke (siehe auch Mindmap 4.1) [CRC02].

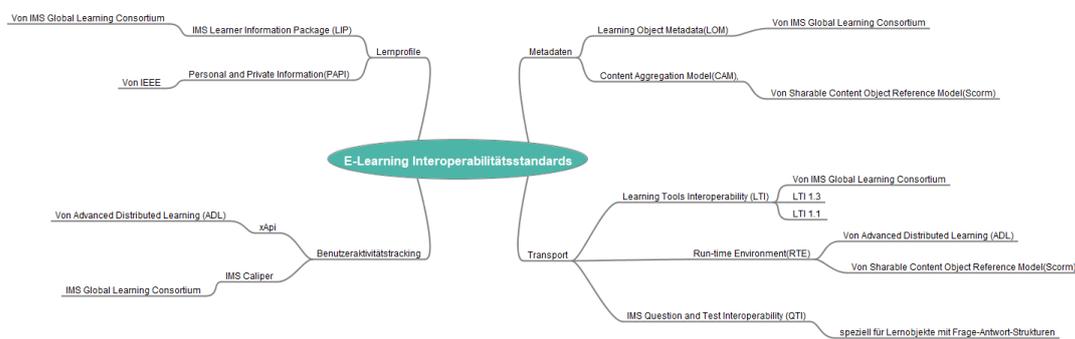


Abbildung 4.1: Einteilung der etablierten Interoperabilitätsstandards in vier Typen: Metadata, Content Packaging, Learner Profile und Content Communication.

Folgende E-Learning-Standards sind in Kategorien eingeteilt [CRC02]

Metadata Hauptbestandteil des E-Learnings sind die Inhalte und Angebote. Für das Verwenden von Lernmaterialien müssen sie einheitlich gekennzeichnet sein, um das Indizieren, Speichern, Ermitteln (Suchen) und Abrufen der Materialien zu unterstützen. Beispielsweise standardisieren folgende Unternehmen Metadaten:

IEEE Learning Technology Standards Committee (LTSC) definieren dafür Learning Object Metadata (LOM) [IEE07].

The Advanced Distributed Learning Initiative (ADL) entwickelten eine Reihe von Richtlinien. Darunter für Metadaten den Content Aggregation Model (CAM), was ein Teil vom Shared Content Object Reference Model (SCORM) ist [The04].

The Dublin Core Metadata Initiative haben allgemein mehrere Standards für Metadaten, unter anderem auch für die Bildung [The].

Content Packaging spezifiziert und standardisiert den Transport der Lerninhalte von einem System zum anderen. Dieser Transport ist notwendig geworden, seitdem Teile vom Kurs

von verschiedenen Personen beispielsweise erstellt, verändert und von verschiedenen Quellen angeboten werden. Das Content Packaging enthält Lernobjekte und zusätzliche Informationen, wie diese zusammengefügt sind. Die Regel der Lieferungen an den Lernenden kann ebenfalls spezifiziert werden.

Folgende Unternehmen beschäftigen sich mit Content-Packaging-Standardisierungen:

IMS Global Learning Consortium entwickelte die Learning-Tools-Interoperability-Spezifikation, die heutzutage am weitesten verbreitet ist [Con].

Dieser Standard wird in 4.3.2 detaillierter erläutert.

IEEE LTSC entwickeln viele Standards, darunter auch ein Standard für den Transport von Lerninhalten [IEE07].

ADL beschreibt dafür die Run-time Environment(RTE) Richtlinie, die ein Teil von SCORM ist [Theo4].

Die Bündelung einer Bewertung und ihrer dazugehörigen Antworten ist eine spezielle Art von Learning Content und wird bei IMS Question and Test Interoperability (QTI) spezifiziert [IMS15].

Learner Profile Ein Learner-Profile-Standard erlaubt das Austauschen von Informationen des Lernenden über verschiedene Systemkomponenten. Informationen können Benutzerdaten, Lernpläne, Lernverlauf, Zugangsanforderungen, Zertifizierung und Rechte, Bewertung der Kenntnisse (Fähigkeit/Kompetenzen) und Status des aktuellen Lernfortschritts sein.

Folgende Spezifikation gibt es für Lernprofile:

Die IMS Learner Information Package (LIP) Spezifikation [IMSb]

Die Personal and Private Information(PAPI) Spezifikation [Alfo6]

Content Communication Um Lerndaten und vorherige Aktivitätsinformationen an die verschiedenen Komponenten der Lernumgebungen zu übermitteln, müssen die Content Communication standardisiert werden. Zum Beispiel umfassen solche Kontexte den Status von Lernaktivitäten oder Ergebnisse bzw. Punktzahlen oder Noten. Standards, die in diese Kategorie gehören sind:

IMS Caliper von IMS Global Learning Consortium [IMSa].

xAPI aka Tin Can API aka Experience API ist eine Weiterentwicklung des SCORM von ADL und weist in vielerlei Hinsicht ähnliche Funktionen mit IMS Caliper auf [GH16]. Dieser Standard wird in 4.3.3 detaillierter erläutert.

4.3.2 Learning Tools Interoperability

Learning Tools Interoperability (LTI) ist ein im Jahr 2010 von IMS Global Learning Consortium entwickelter Standard [Her+10].

Durch diesen Standard ist es möglich, dass das Lernmanagementsysteme eine Vielzahl von externen Funktionen und Diensten miteinander integriert und für den Lernenden eine nahtlose Verwendung anbietet. LTI reduzierte die Implementierungskosten und Wartungen, da diese externen Dienste einfacher eingebunden werden können [Her+10]

Externe Funktionen oder Dienste werden in diesem Kontext als Tool oder Werkzeug bezeichnet. Beispielsweise kann ein Tool ein gehostetes Testsystem oder ein bestimmter Inhalt sein.

Dieser Standard wird von Anbietern für Bildungsinhalten wie z.B. McGraw Hill (ein Lernwissenschaftsunternehmen), der zu den "Big three" der Bildungsverlage zählt [Dav13], verwendet. [Tyl16]. Weitere beliebte Plattformen, die LTI unterstützen, sind Blackboard, LAMS, Moodle, OpenEdx und Schoolbox [Ind19].

Allgemeine LTI-Integration

Die LTI Version 1.0 spezifiziert die zwei verschiedenen Integrationsstile [Her+10].

Basic LTI stellt beim LMS eine Verbindung mit einem Tool auf, die in nur eine Richtung geht.

Dieses Verfahren ist zwar einfach, jedoch begrenzt und es wird nur eine Sicherheitsrichtlinie unterstützt. Das LMS versendet beispielsweise Benutzeridentität, Benutzerrolle und Kursinformationen an das Tool.

Full LTI beinhaltet einen formellen, aushandelbaren Prozess, indem sich die beiden Rollen untereinander einigen, welche Laufzeitdienste sie verwenden. Dies führt zu einer Integration mit einer bidirektionalen Verbindung. Beispielsweise kann das LMS ein Werkzeug wie beim Basic LTI starten und sich zusätzlich Ergebnisse eines Tests zukommen lassen.

Eine allgemeine Architektur eines Netzwerksystems mit Verwendung vom LTI-Standard, die auf einer Serviceorientierten Architektur basiert, sieht folgendermaßen als UML-Komponentendiagramm aus 4.2 (Vergleiche [LQ11]). Das LMS greift mit Hilfe von Full LTI auf ein Tool zu. Dieses Tool besteht aus mehreren Komponenten. Eine der Komponenten ist das Learning Objects Repository. Auf diesem Repository ist das Lernobjekt gespeichert und kann mit einer IMS-Digital-Repositories-Interoperability-Spezifikation abgerufen werden [Kevo3]. Ein anderer Teil umfasst die Evaluation-Engine und ist für die Berechnung der Evaluierung zuständig. Um diese beiden Dienste zu delegieren, wird eine Programming-Exercises-Resolution-Environment-Komponente verwendet. Diese kommuniziert mit dem LMS über LTI.

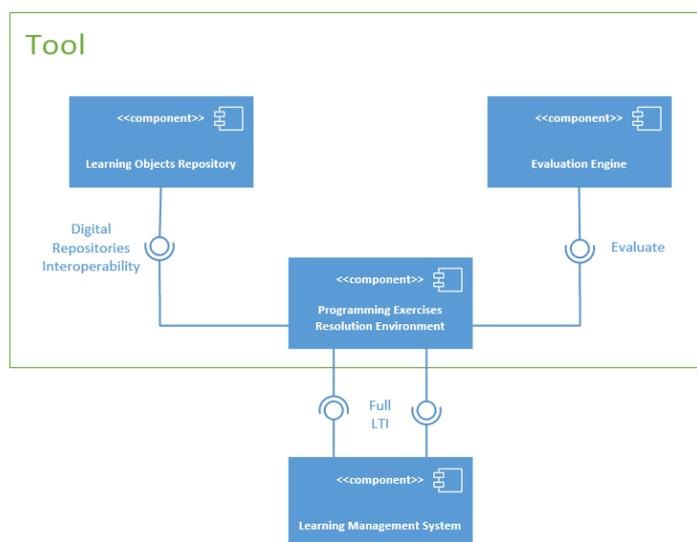


Abbildung 4.2: Allgemeines Komponentendiagramm eines Netzwerksystems mit LTI

Grundlegender Learning-Tools-Interoperability-Aufruf

In diesem Abschnitt werden die Grundlagen des LTI-Aufrufes beschrieben (vergleiche [Her+10]).

Als LTI-Aufruf bezeichnet man den Prozess, bei dem der Benutzer ein externes Tool innerhalb einer Plattform aufruft. Die Daten zwischen Tool und Plattform beim Start werden bei der Integration des Tools in der Plattform im Folgenden definiert [Her+10].

Für die Integration benötigt die Plattform als erstes die Zugangsadresse des Tools, um es ansprechen zu können. Um Zugriff auf die Daten zu erhalten, gibt es drei Varianten, allerdings ist nur eine für externe Werkzeuge relevant und im Folgenden wird nur auf diese eingegangen. Bei dieser Möglichkeit wird ein Key und ein Secret vom Tool in der Plattform konfiguriert. Mit diesen Daten erhält die Plattform den Zugang und verwendet das Secret zum Signieren der Anfrage, damit sicher gegangen werden kann, dass die Anfrage valide ist.

Da alle Plattformen das gleiche Secret für dasselbe Tool verwenden, müssen all diese Plattformen ordnungsgemäß mit den Daten umgehen.

Eine LTI-Anfrage erfolgt über einen HTTP-Post-Aufruf und dabei ist nicht nur eine URL enthalten, sondern auch weitere Daten, die für den Aufruf benötigt werden.

Folgende Parameter sind erforderlich für einen minimalen LTI-Aufruf.

lti_message_type=basic-lti-launch-request beschreibt den Typ der Post-Anfrage, da das Tool verschiedene Anfragetypen unterstützen kann. Hierbei handelt es sich um den Basic-

LTI-Aufruf.

lti_version=LTI-1p0 beschreibt die Version des Standards. Mit '1p0' wird die LTI-Version 1.0 angegeben. Diese Version unterstützt auch LTI 1.1.

resource_link_id = 12098f929-27461 beschreibt, auf welche Ressource der Aufrufer zugreifen möchte. Dafür wird eine eindeutige Identität der Ressource angegeben. In diesem Beispiel ist die Id 12098f929-27461.

Für die Authentifizierung wird der OAuth 1.0 Standard verwendet. Die daraus generierten Parameter sehen folgendermaßen aus.

oauth_consumer_key=b289378-f88d-2929-example.school.edu gibt die Identität des Absenders an, damit das Tool weiß, dass der Absender beispielsweise 'b289378-f88d-2929-example.school.edu' ist.

oauth_signature_method=HMAC-SHA1 gibt an, mit welchem Verfahren die Anfrage signiert ist. In diesem Beispiel wurde es mit einer SHA1-Hashfunktion signiert.

oauth_timestamp=1567618210 zeigt den Zeitstempel an, wann die Anfrage erstellt wurde. Beispielsweise wurde diese Anfrage am Mittwoch, den 4. September 2019 um 19:30 Uhr (GMT+02:00) erstellt.

oauth_version=1.0 beschreibt die Version des OAuth-Standards.

oauth_signature=Xddn2AzBjzwwjgBIVYkvigaKxCdccc%3D ist eine Signierung der gesamten Parameter, damit sicher gegangen werden kann, dass der Aufruf valide ist. Der Aufrufer verwendet dafür eine Hashfunktion und setzt bei dieser Funktion das Secret vom Tool als Parameter.

oauth_nonce=1244834250435893000 und der Zeitstempel sind dafür da, um jede Signierung eindeutig zu identifizieren und aufzupassen, dass jede Nonce nur einmal verwendet wird. Das verhindert, dass Angreifer beispielsweise eine valide Anfrage kopieren können. Der Wert der Nonce ist in der Regel eine zufällig generierte Zeichenkette.

Folgende optionale Parameter können angegeben werden:

user_id=0ae836b9-7fc9-4060-006f-27b2066ac545 identifiziert den Benutzer, der das Tool aufruft.

roles=Instructor beschreibt welche Rolle beziehungsweise welche Rechte der Benutzer hat. Der Instruktor darf Konfigurationen einsehen und verändern. Weitere typische Rollen sind „Lerner“, „Mentor“, „TeachingAssistant“ und „ContentDeveloper“.

lis_person_name_full=Alice B. Cand gibt den vollständigen Namen des Benutzers an.

lis_person_contact_email_primary=user@school.edu gibt die E-Mail-Adresse des Benutzers an.

Weitere optionale Parameter werden in [Her+10] beschrieben.

Wie schon erläutert wurde, verwendet LTI 1.1 das OAuth 1.0a Authentifizierungsframework. Am 16. April 2019 substituierte LTI 1.3 die Authentifizierung von OAuth 1.0 mit einem neuen Sicherheitsmodell für die Nachrichten- und Dienstaauthentifizierung [IMS19]. Dieses Modell verwendet OpenID Connect, JWT-Signature und den OAuth2.0-Workflow (Von Internet Engineering Task Force (IETF)). Die Authentifizierung erfolgt folgendermaßen (vergleiche [IMS19]). Die Plattform fungiert als OpenID-Anbieter und LTI-Nachrichten sind OpenId-Tokens, die die Identität des Nutzers von der Plattform an das Tool kommuniziert. Nachdem sich der Nutzer authentifiziert hat und identifiziert worden ist, wird geprüft, ob der Nutzer die Rechte für den Zugriff auf die Ressourcen oder die Funktionen hat. Für das Erweitern von LTI 1.1 auf LTI 1.3 müssen Plattformen und Tools wie folgt angepasst werden.

Für die LTI-Aufrufsicherheit wird das weit etablierte OAuth2.0 und eine HTTPS-Verschlüsselung über TLS verwendet. Das neue IMS-Sicherheitsmodell erweitert die Sicherheit mit JSON-Web-Tokens (JWT) für die OAuth2-Nachrichtensignatur.

Der Ablauf eines LTI-1.3-Aufrufs wird in einem Sequenzdiagramm in Abbildung 4.3 dargestellt [Mil18].

1. Die Plattform initiiert einen Third-Party-Login, indem eine Anfrage für ein ID-Token an die Login-Tool-URL gesendet wird. Diese Anfrage beinhaltet die Plattform-URL, User login ID, Tool launch URL, Ressourcen Link ID.
2. Das Login-Tool überprüft die Gültigkeit der Anfrage von der Plattform und sendet eine Autorisierungsantwort an die Plattform. Das angefragte ID-Token ist in der URL des Ressourcenlinks.
3. Die Plattform erstellt einen Json-Web-Token (JWT) mit allen LTI-Parametern, die benötigt werden und signiert es mit einem privaten Schlüssel. Anschließend sendet die Plattform eine POST-Anfrage mit dem signierten Token an das Lerninhalt-Tool.
4. Das Lern-Tool fordert vom Public Key Tool den öffentlichen Schlüssel der Plattform an und überprüft die Gültigkeit der Signierung des JWTs und sendet den Inhalt an die Plattform, welcher dann im Browser angezeigt wird.

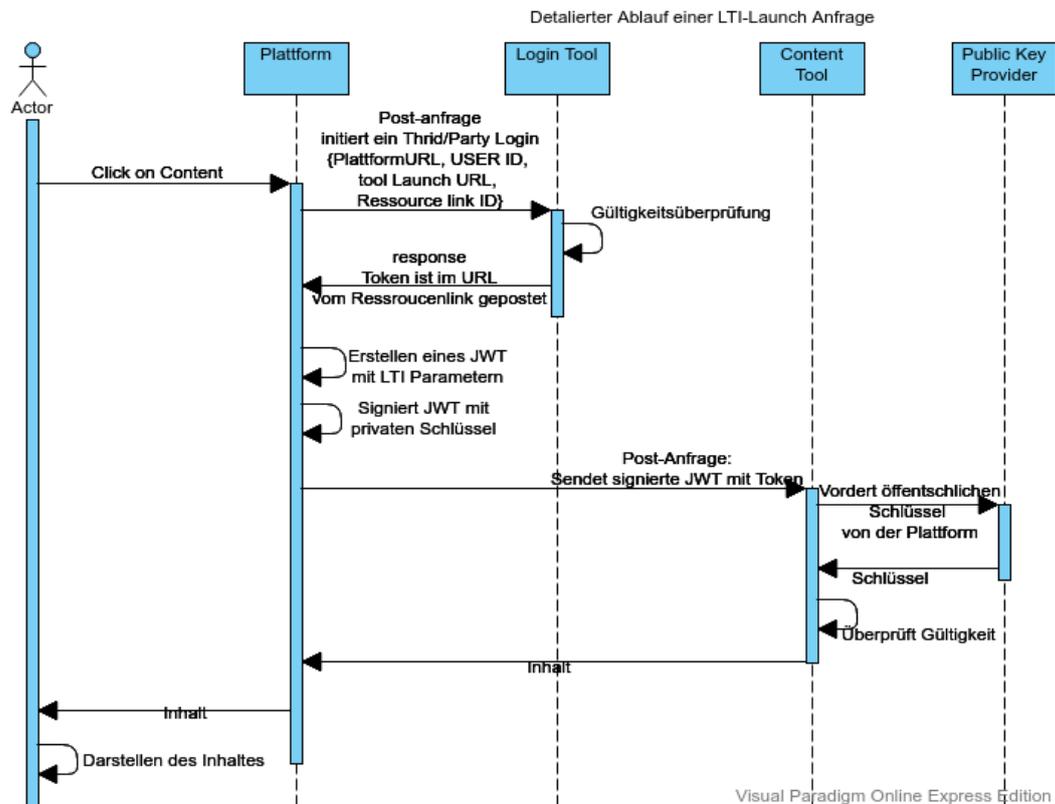


Abbildung 4.3: Sequenzdiagramm eines LTI 1.3 Aufrufes

4.3.3 xAPI

Die Experience API, oder xAPI ist eine Spezifikation für Lerntechnologien, die es ermöglicht, Daten über die Lernaktivitäten einer Person zu sammeln und auszutauschen [Sof]. Die xAPI beschreibt die Aktivitäten in einem einheitlichen „Subjekt-Verb-Objekt“-Format. Die Notation ist in JSON und kann folgendermaßen aussehen.

```

1 {
2   "actor": {
3     "name": "Max Mustermann",
4     "mbox": "mailto:max.mustermann@example.com"
5   },
6   "verb": {
7     "id": "http://adlnet.gov/expapi/verbs/experienced",
8     "display": { "en-US": "experienced" }
9   },
10  "object": {

```

```
11     "id": "http://example.com/activities/solo-hang-gliding",
12     "definition": {
13         "name": { "en-US": "Solo Hang Gliding" }
14     }
15 }
16 }
```

Die xAPI bietet für sehr unterschiedliche Systeme die Möglichkeit, miteinander zu kommunizieren, indem die Aktivitäten in einem einfachen und einheitlichen Wortschatz dargestellt werden. Die Statements einer Aktivität werden in einem Datenbank-ähnlichen Learning Record Tool archiviert und können allein existieren oder sind in einer Lernplattform platziert.

Durch den Aufbau eines xAPI-Statements können dadurch extrem viele unterschiedliche Statements festgehalten werden. Dabei handelt es sich um die größte Stärke, allerdings auch um die größte Schwäche von xAPI. Denn Mehrdeutigkeit ist möglich, da die Vokabeln ebenfalls Synonyme besitzen. Zum Beispiel gibt es im Englischen das Wort für „abgeschlossen“ als finished, completed, concluded, accomplished, etc. [Bow13]. Um dieses Problem zu kompensieren, wird ein xAPI-Vokabular und Semantik angeboten, in denen die Bedeutung und Anwendung der Wörter definiert werden [W3C].

4.4 Unity

Unity ist eine Game Engine, d.h. eine Laufzeit- und Entwicklungsumgebung für Videospiele [Unib]. Im Folgenden wird kurz die Benutzeroberfläche und die allgemeine Verwendung der Game Engine beschrieben. In der Abbildung 4.4 ist die Benutzeroberfläche ersichtlich. In der Mitte befindet sich das Hauptfenster (rot markiert) und stellt die Szene in 2D oder wie hier in 3D dar. In dieser Abbildung ist eine Szenenkamera zu sehen, die die Ansicht des Benutzers anzeigt. Weitere Objekte in der Szene sind drei verschiedene 3D-Objekte. Eine Szene ist im „Hierarchy“-Fenster (blau markiert) als Szenengraph aus sogenannten „GameObjects“ organisiert. Hier können auf alle Objekte in der Szene zugegriffen werden. Game-Objekte können mit Werkzeugen (gelb markiert) skaliert, verschoben oder gedreht werden. Die Transformation kann man ebenfalls beim „Inspector“-Fenster nutzen (grün markiert), wenn man das Objekt auswählt und die gewünschte Transformationen manuell bei „Transform“ angibt. Hier können dem ausgewählten GameObject zusätzliche Komponenten, wie beispielsweise Materialien, Klänge, physikalische Eigenschaften oder Skripte hinzugefügt werden. Bei der „Game“-Registerkarte werden grafische Darstellung und Verhalten des Spiels simuliert, indem man die Navigationsknöpfe betätigt (türkis markiert). Im „Project“-Fenster (orange markiert) werden die Ressourcen für das gesamte Projekt organisiert.

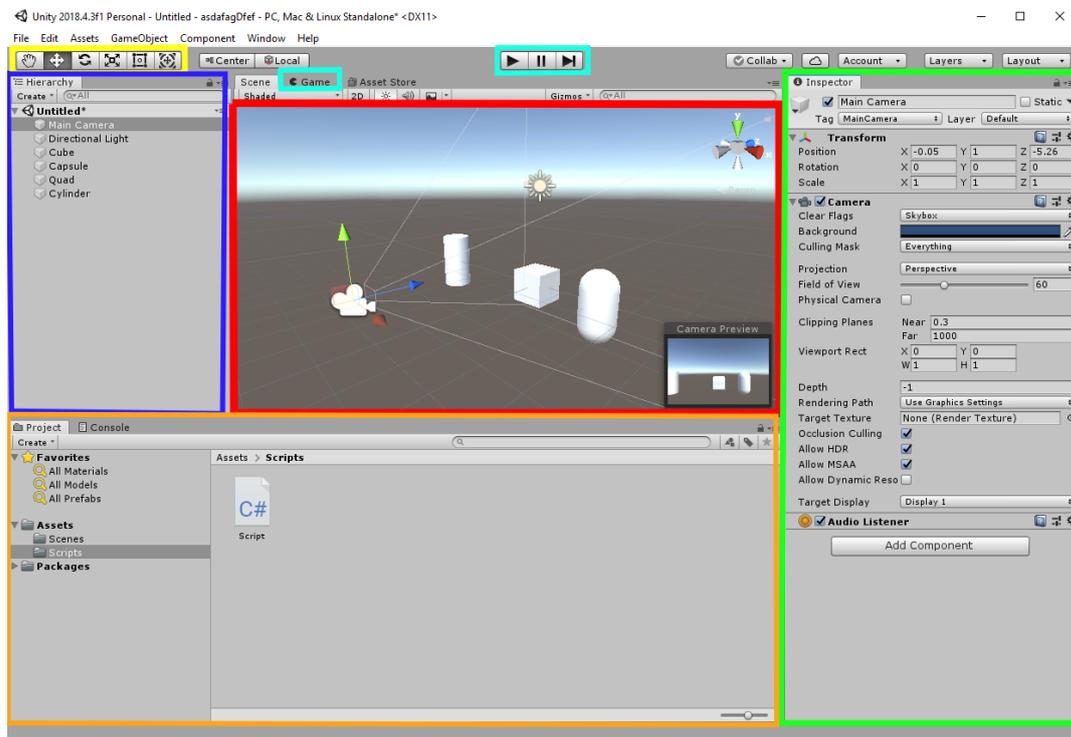


Abbildung 4.4: Die Unity Game Engine Entwicklungsumgebung

5 Stand der Forschung und Technik

In diesem Kapitel werden die bisherigen Entwicklungen der Kopplung von Lernumgebungen mit dem Einsatz von Interoperabilitätsstandards untersucht. Zuerst wird kurz die Kopplung von verschiedenen Lernmanagementsystemen betrachtet und anschließend die Kopplung von Lernmanagementsystemen mit externen Applikationen analysiert.

5.1 Kopplung von Lernmanagementsystemen

Declan Dagger et al. beschreibt 2007 eine feingranulare Serviceorientierte Architektur für die Lernumgebungen. Dadurch ist es möglich, die Umgebung in LMS und Lerninhalt zu trennen. Somit ist die Umgebung nicht mehr monolithisch. Die Inhalte werden durch die Architektur exportierbar und importierbar gemacht. Zusätzlich können die heterogenen Umgebungen die Inhalte untereinander austauschen. Beispielsweise können Metadaten zwischen verschiedenen Umgebungen ausgetauscht werden [Dag+07]. Genau das wurde 2010 mit dem LTI-Standard einheitlich spezifiziert [Her+10]. Moderne Lernmanagementsysteme übernehmen diese Architektur bzw. die LTI-Spezifikation [Ind19].

5.2 Kopplung von Lernplattformen mit Lernspielen

Die Arbeit „Using e-learning standards in educational video games“ von Ángel del Blanco et al. beschreibt, wie E-Learning-Standards genutzt werden, um die Integration von Lernspielen in Lernmanagementsysteme zu verbessern [Del+13]. Des weiteren wird eine eigene Umsetzung einer Lernplattform mit einem Lernspiel vorgestellt. In dieser Arbeit werden ältere Interoperabilitäten wie SCORM und IMS CC verwendet und drei verschiedene Integrationsarten eines Lernspiels im LMS beleuchtet (siehe Abbildung 5.1).

Modell a) Bei einem „Black-Box“-Modell stellt die Plattform einen Lerninhalt-Service bereit und darin befindet sich das Spiel. Der Lerninhalt-Service ist lediglich dafür da, das Spiel zu starten. Die Plattform hat keinerlei Informationen, was der Lernende im Spiel macht.

Modell b) Das „White Box“-Modell ähnelt dem Black-Box-Modell, denn das Spiel wird ebenfalls von der Plattform angeboten. Der Unterschied ist, dass eine aktive Kommunikation mit der Lernplattform stattfindet, beispielsweise mit einem „Studenten Profile Service“.

Modell c) Bei einem „Decoupled“-Modell wird das Spiel entkoppelt über einen Proxy im Lerninhalt-Service angeboten. Das Spiel existiert extern auf einer Spieleplattform und kommuniziert mit Services von der Spieleplattform.

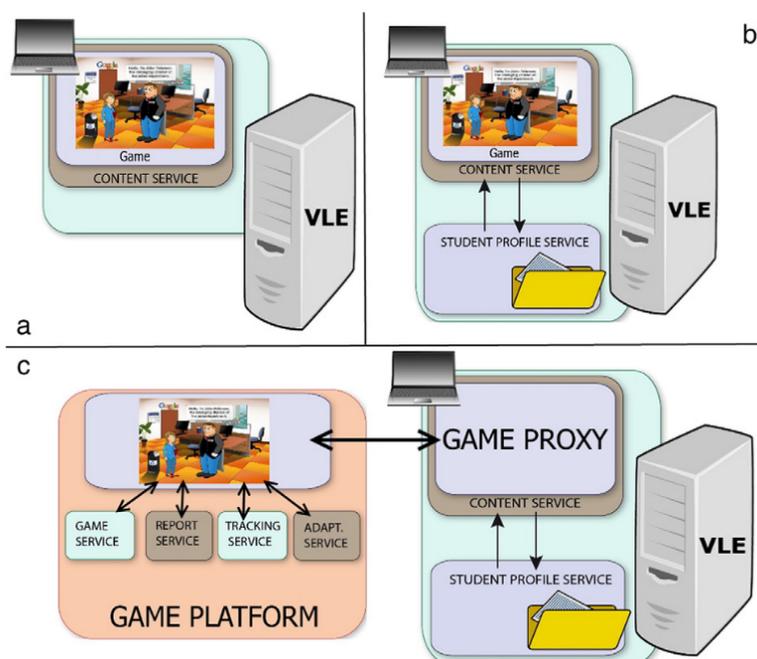


Abbildung 5.1: Drei verschiedene Arten der Integration eines Lernspiels in ein Lernmanagementsystem. Bei der Integration a) handelt sich um ein „Black-Box“-Modell, bei b) um ein „White Box“-Modell and bei c) um ein „Decoupled“-Modell (Bild aus [Del+13]).

5.3 Unity und xAPI

Global xAPI veröffentlichte im August 2018 einen Blog mit dem Titel „Getting Started with a Unity 3D xAPI Pilot“. In diesem Blog kündigten sie ein xAPI-Framework für Unity [Cla18] an. Allerdings ist dieser noch in der Entwicklung. Sie entwickelten zur Demonstration einen webbasiertes Spiel. Dieses Spiel erfasst Benutzeraktivitäten und speichert sie auf einem externen LRS. Man kann die Demo auf der Seite ausprobieren. Außerdem können auch auf die xAPI-Statements vom Blog zugegriffen werden.

6 Konzeption

In diesem Kapitel werden die Anforderungen für die Entwicklung definiert und näher betrachtet. Anschließend wird ein Grundkonzept erstellt und eine Architektur aufgestellt.

Wie in der Zielsetzung in 3.2 beschrieben, ist das Ziel dieser Bachelorarbeit eine webbasierte Lernplattform mit einem Serious Game zu koppeln, welches zusätzlich externe Lernmaterialien anbietet. Aus dieser Zielsetzung wird ein grober Rahmen der Konzeption gegeben.

6.1 Anforderungsanalyse

Als erstes muss klargestellt werden, welche Funktionen schon bereits angeboten werden und welche Funktionen implementiert werden müssen, um anschließend die Anforderungen dieser Arbeit zu analysieren.

6.1.1 Überblick

Momentan bietet die CLM eine Lernplattform an, mit der man auf einen externen Computer, auf dem das Lernspiel LE läuft zugreift (via Remote-Desktop-Windows-Applikation). Der Lernende kann dadurch interaktiv das Serious Game bedienen. Wenn der Lernende externe Lerninhalte aufruft, wird das Spiel minimiert, ein Browser mit dem Link zum Inhalt gestartet und der Lernende muss sich erneut anmelden, um den Lerninhalt einzusehen. An diesem Standpunkt knüpft diese Bachelorarbeit an und verfolgt folglich das Ziel, die Lerninhalte der Fraunhofer CLM zu erweitern. Dafür soll das eingebundene existierende Lernspiel LE zusätzliche Lerninhalte anbieten, um einen reibungslosen Lernprozess zu ermöglichen.

Das Aktivitätsdiagramm in der Abbildung 6.1 verschafft einen Gesamtüberblick über die Aktivität eines Lernenden und die zu implementierenden Funktionen. Wie zuvor beschrieben, verwendet der Lernende wie gewohnt die Anwendung. Bei der neuen Funktion kann der LE-Nutzer im Spiel externe Inhalte aufrufen und anzeigen. Anschließend kann er sich den Lernstoff aneignen und beispielsweise Prüfungen ablegen. Bei einer Vervollständigung eines Kurses oder Tests erhält der Lernende im Spiel eine Bewertung und ein Feedback. Wenn der Benutzer fertig mit dem Lernen ist, beendet er das Spiel und schlussendlich die Remote-Desktop-Applikation.

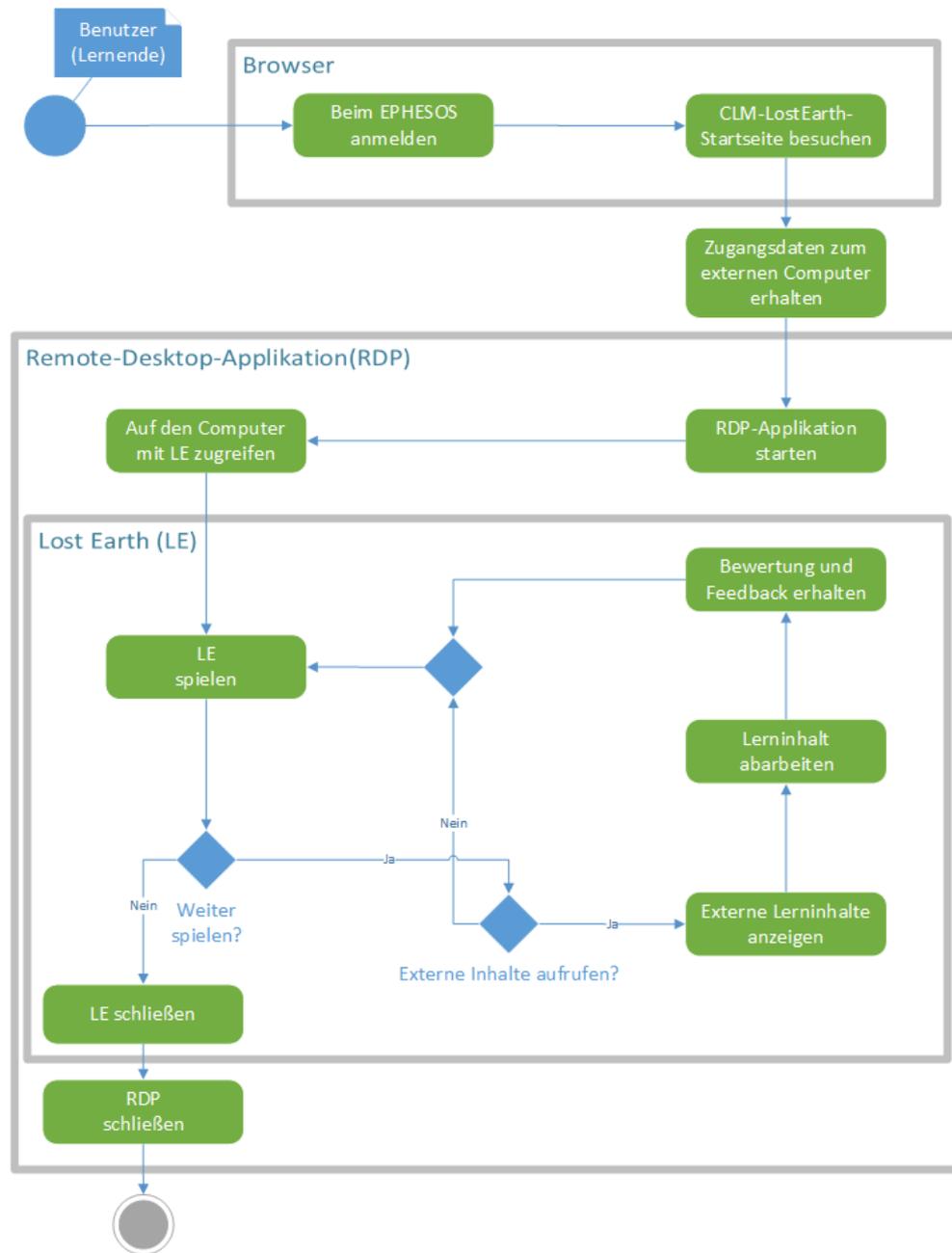


Abbildung 6.1: Aktivitätsdiagramm der allgemeinen Anforderungen

6.1.2 Anforderungen

Die Anforderungen, die in dieser Bachelorthesis umgesetzt wurden, werden aus dem Aktivitätsdiagramm 6.1 Schritt für Schritt analysiert und in diesem Unterkapitel erläutert.

- **Übertragung von Benutzerdaten an das LE:**
Da das LE aufgrund der Remote-Desktop-Applikation vollständig abgeschottet ist und keinerlei Informationen über den Benutzer hat, ist es notwendig, Benutzerinformationen von der CLM-Lost-Earth-Webseite an das Serious Game LE zu schicken.
- **Anbieten der externen Lernmaterialien für das LE:**
Um überhaupt Lernmaterialien abrufbar zu machen, müssen die Lernmaterialien über das Internet angeboten werden.
- **Authentifizierung für die externen Lerninhalte verläuft im Hintergrund:**
Um den Lernprozess so reibungslos wie möglich zu gestalten, muss die Authentifizierung im Hintergrund erfolgen, ohne dass sich der Benutzer zusätzlich anmelden muss.
- **Anzeigen und Interagieren mit den externen Lerninhalten:**
Dies erfolgt im Spiel und muss eine flexible Benutzerschnittstelle sein, um so skalierbare Lerninhalte anbieten zu können.
- **Kurs oder Test abschließen:**
Um den Lernerfolg kontrollieren zu können, muss das Abschließen von Kursen oder Tests bewertet werden.
- **Einfluss der Ergebnisse im Lost Earth:**
Um die Bemühung eines guten Abschlusses eines Tests oder Kurses zu erhöhen, muss das Ergebnis je nach Abschluss einen positiven oder negativen Effekt auf den Spielverlauf auswirken.
- **Übertragung der Ergebnisse an das LE:**
Um überhaupt den Spielverlauf mit dem Ergebnis eines Kurses oder Tests beeinflussen zu können, muss das Ergebnis an das LE übermittelt werden.
- **Übertragung von LE-Benutzerinteraktionen:**
Um andere Lernsysteme über die Lernfortschritte im Spiel informieren zu können, müssen die Benutzerinteraktionen erfasst und extern gespeichert werden.

6.2 Bestimmen der Komponenten und Interoperabilitätsstandards

In diesem Kapitel wird die Dienste bzw. die Komponenten festgelegt. Dafür werden die Aufgaben in einzelne Komponente zugeordnet und näher beleuchtet. Anschließend werden Kommunika-

tionsstandards zwischen den Komponenten bestimmt und mithilfe von Sequenzdiagramme den gesamten Lernprozess des Benutzers aufzeichnet. Zu Letzt wird die Gesamtarchitektur und ihren Eigenschaften vorgestellt

6.2.1 Komponenten

In diesem Unterkapitel werden folglich die Komponenten aus dem Aktivitätsdiagramm 6.1 und den Anforderungen 6.1.2 ermittelt und beleuchtet.

- **Fraunhofer CLM:**

Die Middleware CLM vom Fraunhofer IOSB bietet das **Anmeldeportal EPHEOS** an und darüber hinaus ein **Lost-Earth-Launcher-Tool**, das die Zugangsdaten zum externen Computer mit LE verwaltet und die Benutzerdaten an ein Benutzerdaten-Tool sendet, um diese für die LE-Applikation bereitzustellen.

- **Tool für das Managen der Benutzerdaten:**

Das Benutzerdaten-Tool ist ein einfaches Tool im Netz, das vom CLM die Benutzerdaten erhält, temporär speichert und bei Anfrage vom LE diese Daten weiter versendet. Diese Komponente zwischen CLM und LE ist notwendig, da das LE in der Remote-Desktop-Umgebung vollständig abgeschottet ist und keinerlei Benutzerinformation besitzt.

- **Lost Earth:**

Das Serious Game Lost Earth (erläutert in 4.2) bietet eine Lernumgebung mit internen und externen Inhalten an.

- **Tool für externe Lernmaterialien:**

Hierbei handelt es sich um ein Tool, das Lerninhalte für externe Lernsysteme anbietet. Bei exklusiven Lerninhalten ist ein Authentifizierungsmechanismus notwendig.

- **Eingebetteter Browser:**

Um im LE flexible und vielfältige Lerninhalte anzubieten, ist ein eingebetteter Browser im Spiel optimal, da dieser die gleichen Funktionen wie ein normaler Browser anbietet.

- **Tool für die Lernaktivitäten:**

Dieses Tool wird für die Kommunikation der Lernaktivitäten zwischen verschiedenen Lernsysteme benötigt, da beispielsweise die Lernumgebung LE und die Lernplattformen Benutzertätigkeiten austauschen, welche Aktivitäten der Lernende im LE verrichtet hat. Damit wird der Standpunkt des Lernfortschrittes gesichert. Beispielsweise möchte man einen Kurs, den man abgeschlossen hat, nicht noch einmal erledigen müssen. Dieses

Tool verwaltet dafür eine Art Datenbank, auf der das Tool die Lernaktivitäten speichert und abrufen.

6.2.2 Interoperabilitätsstandards

Aus den vorherigen Unterkapiteln, kristallisiert sich heraus, welche Interoperabilitätsstandards, für den Anforderungsfall geeignet sind. Dafür wird folglich die Kommunikation der einzelnen Komponente betrachtet.

Folgende Kommunikation ist realisiert

„*Fraunhofer CLM*“ → „*CLM-Lost-Earth-Startseite*“

Die Plattform ruft die CLM-Lost-Earth-Startseite mit LTI-Standard auf und ist somit ein LTI-Tool.

Folgende Kommunikationen wurden umgesetzt

„*CLM-Lost-Earth-Tool*“ → „*Benutzerdaten-Tool*“

Bei dieser Kommunikation werden die Nutzerdaten des Lernenden an den Benutzerdaten-Tool gesendet. Verwendet wird dafür der LTI-Standard, da die Integration eines LTI-Tool vom der CLM unterstützt wird. Somit ist das Benutzerdaten-Tool ein LTI-Tool. Ferner kann die Benutzerdaten als Parameter im LTI-Aufruf angegeben werden.

„*Lost Earth*“ → „*Benutzerdaten-Tool*“

Da das Benutzerdaten-Tool ein LTI-Tool ist, werden die Benutzerdaten über ein LTI-Aufruf geholt.

„*Eingebetteter Browser*“ → „*Tool für externe Lernmaterialien*“

Für das Integrieren der externen Funktionen und Dienstleistungen bietet sich der LTI-Standard an. Daraus folgert, dass der Tool für externe Lernmaterialien ein LTI-Tool sein muss.

„*Lost Earth*“ → „*Eingebetteter Browser*“

LE erhält die Benutzerdaten vom Benutzerdaten-Tool und diese Daten werden an den Browser weiterleitet muss. Dafür wird kein Standard benötigt, da die beiden Komponente in einer Anwendungsumgebung befinden.

„*Tool für externe Lernmaterialien*“ → „*Tool für Lernaktivitäten*“

Das Tool für externe Lerninhalte sendet Benutzeraktivitäten an den Lernaktivitäten-Tool, um diese Information für externe Lernumgebungen bereit zu stellen. Für diese

Anwendung eignet sich der xAPI- oder IMS-Caliper-Standard, da diese genau für solche Anwendungen entwickelt wurde. Für die Umsetzung wird xAPI-Standard verwendet, da der von CLM und weitere Systeme unterstützt werden. Beim xAPI werden Lernaktivitäten als xAPI-Statements bezeichnet und für das Abrufen und das Archivieren dieser Statements wird ein *Learning Record Store(LRS)* benötigt.

„Lost Earth“ → „Tool für Lernaktivitäten(LRS)“

Da LE beispielsweise Testergebnisse erfahren möchte, die beim externen Lernmaterialien erbracht wurden, ruf das Spiel die Statements des Lernenden vom LRS auf. Außerdem sendet LE in xAPI Standard Benutzeraktivitäten an den LRS.

6.3 Kompletter Anwendungsablauf

Für ein besseres und detailliertes Verständnis des gesamten Anwendungsablaufes werden dafür vier Anwendungsfälle mithilfe von Sequenzdiagrammen aufgezeichnet. Bei den folgenden Fällen ist die Zusammenarbeit der einzelnen Komponente näher zu betrachten.

1. Use Case: Lernspiel Lost Earth starten (Abbildung 6.2)

Der Lernende meldet sich mit seinem Benutzernamen und Passwort bei der Ephesos-Plattform an. Die Plattform verifiziert den Login als gültig und gibt dem Lernenden Zugriff auf die Lernangebote des Ephesos. Der Nutzer klickt auf den LE-Kurs, um auf die Startseite des Kurses zu gelangen. Für die Anzeige der LE-Startseite, stellt die Plattform eine LTI-Anfrage an den Lost-Earth-Tool, während das Tool die Benutzerinformationen an das Benutzerdaten-Tool schickt. Anschließend klickt der Nutzer auf den Startknopf der LE-Startseite und erhält daraufhin die Zugangsdaten des RDP. Der Benutzer startet die native Windows-RDP-Applikation, um auf den externen Computer zu zuzugreifen, indem er Name und Passwort eingibt. Der Benutzer beginnt das Serious Game. Da aber das Spiel keine Informationen vom Lernenden besitzt, wird eine Anfrage an das Benutzerdaten-Tool gesendet, das mit den Nutzerdaten antwortet und das Spiel kann schließlich den Lernenden identifizieren.

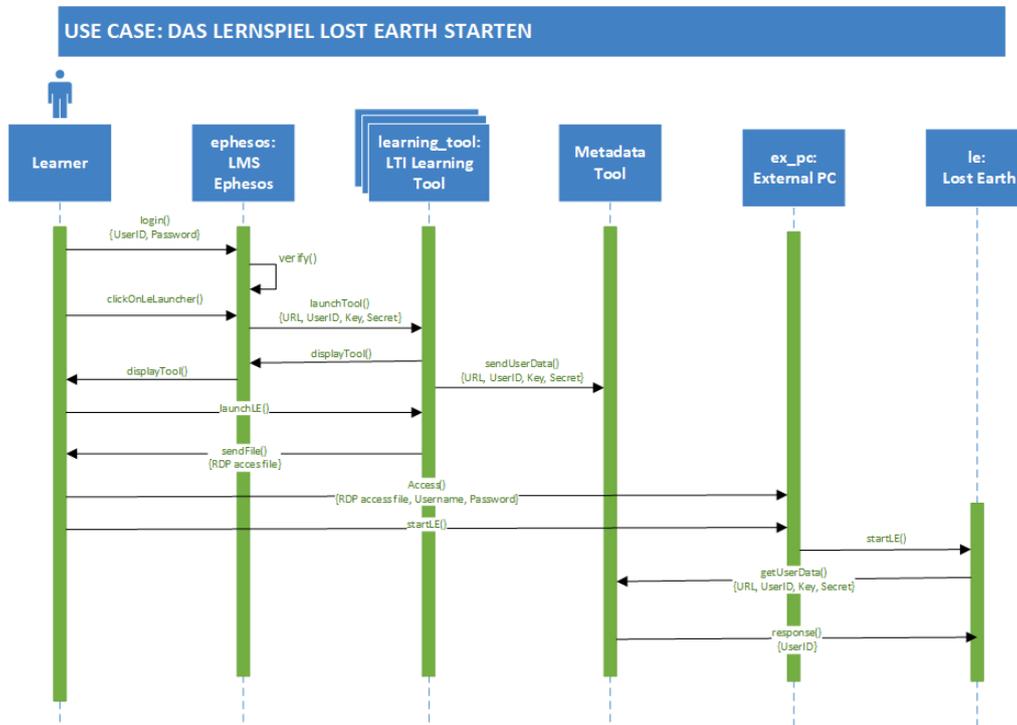


Abbildung 6.2: Veranschaulicht den Ablauf des Benutzers, wenn er das Lost Earth Lernspiel startet und zeigt auf wie die Komponente miteinander kommunizieren. Zum Starten wird die Webstartseite vom CLM-LE angezeigt und der Lernende erhält einen RDP-Zugang zum externen PC, von dem er das LE starten kann.

2. Use Case: Externe Lerninhalte im Lost Earth aufrufen (Abbildung 6.3)

In diesem Anwendungsfall hat sich der Lernende erfolgreich den Zugriff zum Computer verschafft und lernt im LE-Universum. Nun möchte der Lernende externe Lerninhalte aufrufen. Dafür öffnet der Lernende im Spiel den eingebetteten Browser. Der Browser erhält die Nutzerdaten vom LE und signiert die LTI-Anfrage mit dem Secret vom Lern-Tool. Der Browser sendet die signierte Anfrage an das externe Lern-Tool, womit die URL des Lern-Tools, Benutzerdaten, das Key sowie die Signierung als Parameter mitgeschickt werden (Siehe für weitere Parameter in Grundlagen der LTI-Aufruf bei 4.3.2). Das LTI-Tool für Lerninhalte verifiziert den Aufruf und holt vom LRS gegebenenfalls xAPI-Statements des Benutzers. Das LTI-Tool richtet für den Lernenden die Seite ein: Beispielsweise werden erledigte Aufgaben als abgeschlossen angezeigt. Das Tool sendet die eingerichtete Seite als Rückgabewert an den Browser zurück und wird dort angezeigt.

USE CASE: EXTERNE LERNINHALTE IM LOST EARTH AUFRUFEN

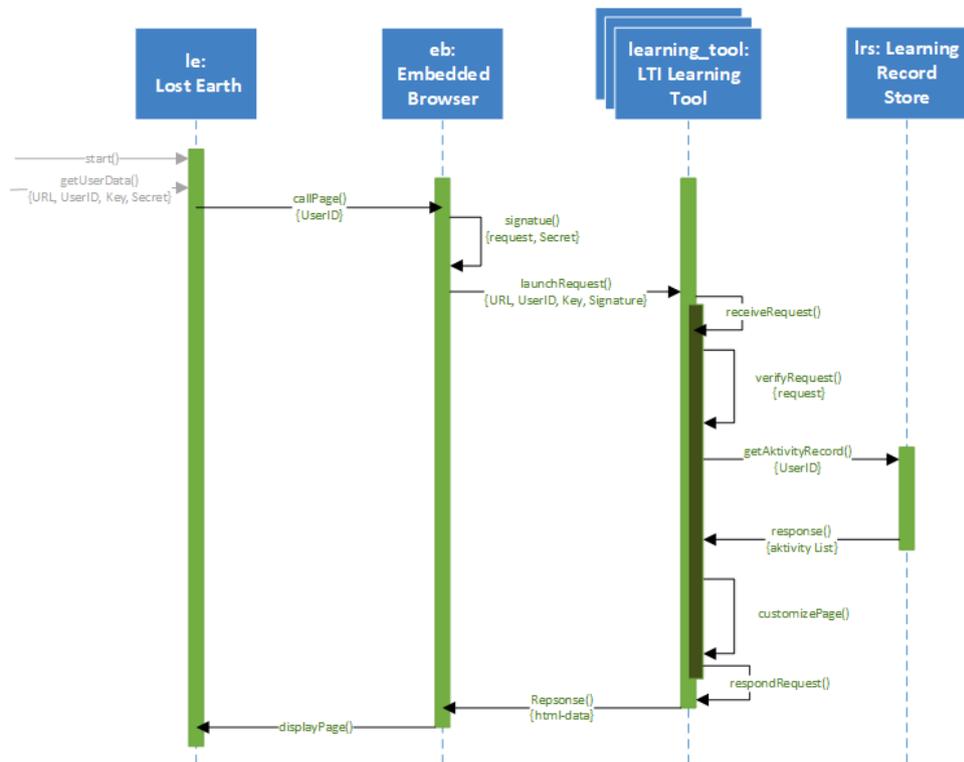


Abbildung 6.3: Veranschaulicht den Ablauf des Benutzers, wenn er Lerninhalte im LE aufruft. Dafür wird der interne Browser geöffnet und eine LTI-Anfrage an das Lernmaterial-Tool sendet.

3. Use Case: Test im Browser bearbeiten (Abbildung 6.4)

In diesem Anwendungsfall bearbeitet der Lernende einen Test oder eine Aufgabe im Browser. Wenn der Nutzer eine Antwort gibt oder eine relevante Aktivität macht, informiert der Browser das Lernmaterial-Tool, indem er das Statement sendet. Das Lern-Tool sendet das Statement weiter an der LRS, das das Statement in ein LRS speichert, und gibt schließlich dem Browser eine Rückmeldung.

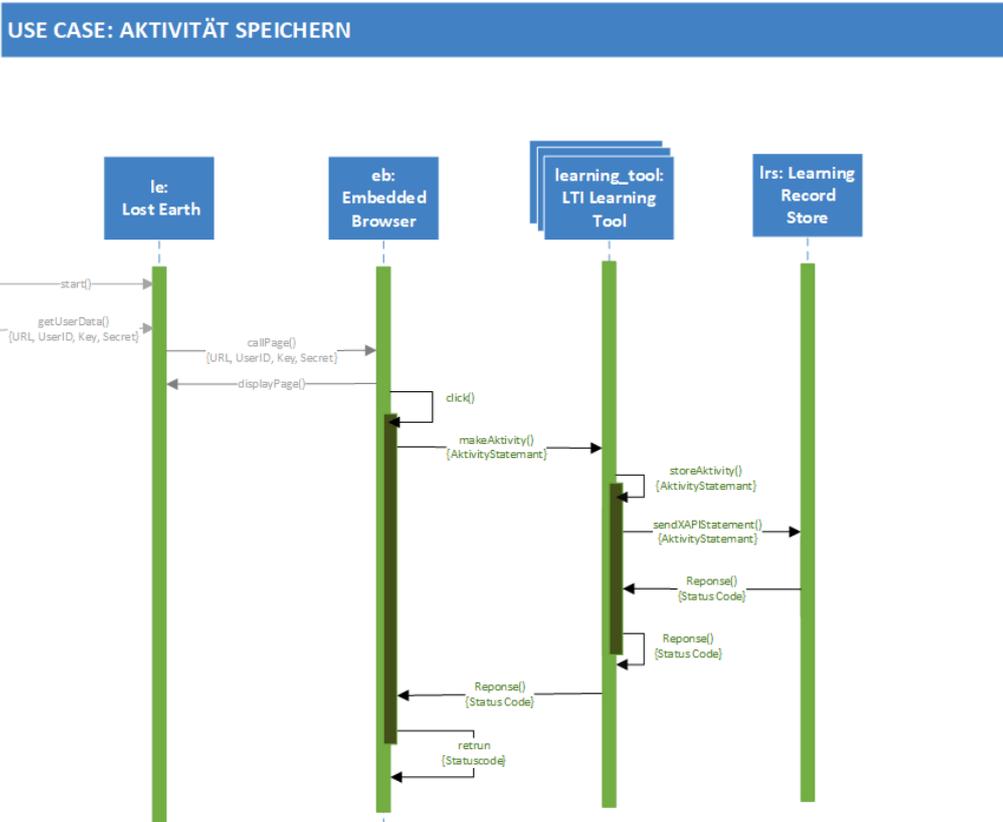


Abbildung 6.4: Veranschaulicht den Ablauf des Benutzers, wenn er im Browser ein Statement macht. Dieses Statement wird an den Lern-Tool gesendet und wird anschließend an den LRS weitergeleitet, um es dort zu archiviert.

4. Use Case: Aufgabe abgeben und bewerten lassen (Abbildung 6.5)

In diesem Anwendungsfall lässt sich der Lernende prüfen und sendet die Antworten an das Lern-Tool, das die Antwort mit den Lösungen vergleicht und die Wertung daraus berechnet (siehe auch Tool-Komponente in 4.2). Das Tool speichert diese Wertung als xAPI-Statement im LRS. Danach sendet das Lern-Tool ein Feedback, das im Browser angezeigt wird. Das LE ruft anschließend die Wertung vom LRS ab und ändert anhand dessen den Spielverlauf. Bei einem schlechten Ergebnis werden beispielsweise Ressourcen im Spiel abgezogen.

USE CASE: AUFGABE BEWERTEN

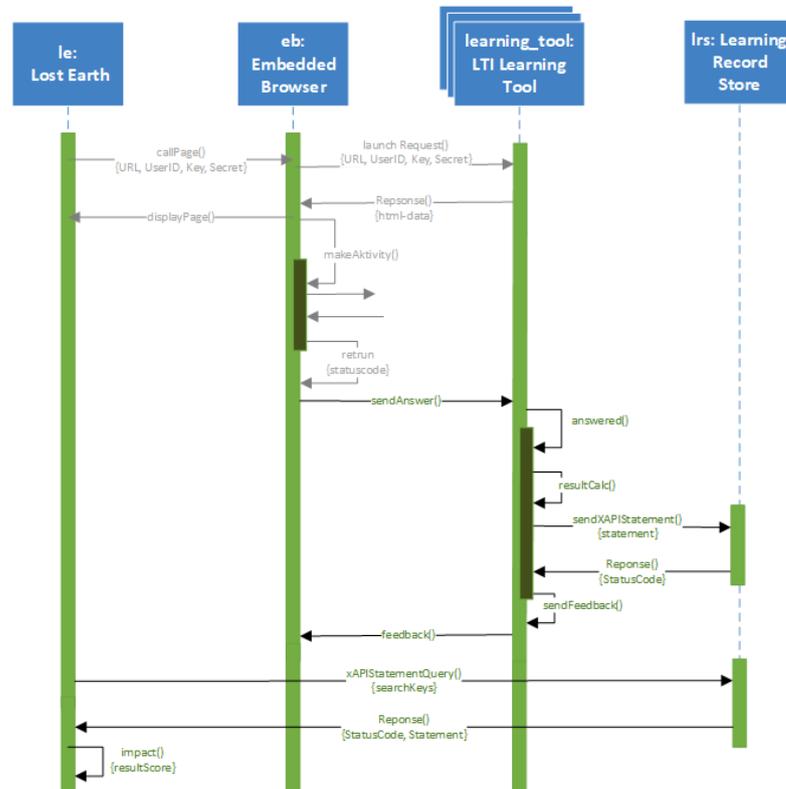


Abbildung 6.5: Veranschaulicht den Ablauf des Benutzers, wenn er eine Aufgabe oder einen Test abgibt. Das Lern-Tool erhält die Antwort des Nutzers und berechnet daraus das Ergebnis. Das Resultat wird im LRS gespeichert und im Browser angezeigt. Das LE holt sich das Ergebnis vom LRS und ändert anhand dessen den Spielverlauf.

5. Use Case: LE sendet xAPI-Statements (Abbildung 6.6)

Nach dem erfolgreichen Starten des LE, sendet das Spiel einen LE-Start-Statement an den LRS. Der Lernende lernt solange bis er nicht mehr möchte und möchte das Spiel schließen. Er drückt auf das Beendenknopf und dabei wird noch ein xAPI-Statement an den LRS gesendet und speichert es auf seinen Speicher.

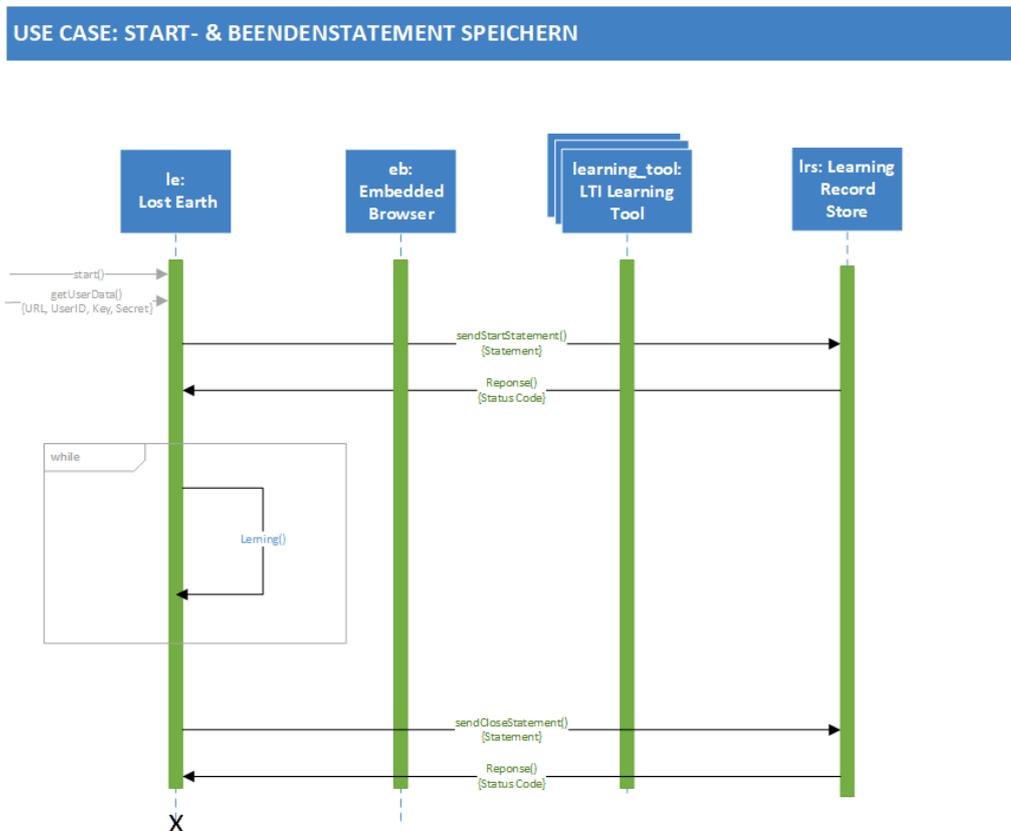


Abbildung 6.6: Veranschaulicht den Ablauf des Benutzers, wenn er das Spiel startet und schließt. Dabei wird ein xAPI-Statement an den LRS gesendet.

6.4 Architektur

Aus den Sequenzdiagrammen im vorherigen Kapitel 6.3 leitet sich die Gesamtarchitektur ab (siehe Abbildung 6.7). In dieser Architektur wird nochmal die gesamte Kopplung ersichtlich. Die LTI-Tools Lost-Earth-Launcher-, Benutzerdaten und Lernmaterial-Tool werden mit LTI-HTTP-Post-Anfrage erreicht und das LRS wird über die Rest-API kommuniziert. Bei der Architektur handelt es sich um eine Serviceorientierte Architektur. Das LE nutzt lose und verteilte Dienste hier als Tools bezeichnet, um eine flexible und wiederverwendbare Lernumgebung zu schaffen. Die externen Lernmaterialien im LE kann man beliebig erweitert und auszutauschen, da das Spiel fähig ist alle LTI-Tools aufzurufen. Beispielsweise können andere Lernumgebungen die Lerninhalte im Tool aufrufen, wenn sie die notwendigen Zugangsdaten besitzen.

7 Implementierung

Bei der Implementierung wird die agile Projektmanagementmethode SCRUM verwendet. Bei dieser handelt es sich um einen inkrementellen und iterativen Entwicklungsprozess. Dazu werden die Anforderungen aus Kapitel 6.1.2 priorisiert und inkrementell implementiert. Bei jeder Iteration (Sprint) werden Anforderungen nach ihrer Priorität ausgesucht, diese implementiert und somit das Produkt nach und nach erweitert.

Da das Serious Game LE ein Unity Spiel ist, wird dafür die Unity-Entwicklungsumgebung verwendet. Die Umgebung wird kurz bei Grundlagen im Abschnitt 4.4 erläutert.

In den folgenden Kapiteln werden die Anforderungen nach ihrer Priorität sortiert und anschließend Schritt für Schritt umgesetzt. Für die Umsetzung wird zuerst ein Prototyp implementiert und erst dann in LE integriert.

7.1 Priorisierung der Anforderungen

Die Priorisierung dient dafür, dass zuerst die wichtigsten und zum Ende die weniger relevanten Funktionen umgesetzt werden. Für den Überblick wird im Folgenden die Priorisierung als Liste dargestellt.

1. Priorität
 - Anzeigen der externen Lerninhalte
2. Priorität
 - Authentifizierung für die externen Lerninhalte verläuft im Hintergrund
 - Interagieren mit den externen Lerninhalten
3. Priorität
 - Übertragung von LE-Benutzerinteraktionen
 - Übertragung der Ergebnisse an das LE
4. Priorität

- Übertragung von Benutzerdaten an das LE

5. Priorität

- Anbieten der externen Lernmaterialien für das LE

6. – Einfluss der Ergebnisse im LE

Die höchste Priorisierung ist die Kopplung zwischen LE und externen Lern-Tools, da das Aufrufen, Anzeigen der Lernmaterialien und die Interaktion mit diesen die Hauptfunktionen dieser Arbeit darstellt. Diese Funktionen besetzen die ersten zwei Plätze. Darauf folgen die Anforderungen „Kopplung mit xAPI bzw. mit dem Learning Record Store“. Diese Anforderungen haben ebenfalls eine hohe Priorität, weil sie für die Kommunikation zwischen LE und den Lern-Tools benötigt werden. Platz vier wird von der Anforderung „Übertragung von Benutzerdaten an das LE“ besetzt. Dies ist ebenfalls relevant, da sie die Kommunikation zwischen CLM-Lernplattform und LE umsetzt. Weniger relevant ist das Anbieten der Lernmaterialien, da sie nur zur Vorführung dient und theoretisch jedes andere LTI-Tool verwendet werden kann. Die Anforderungen „Einfluss der Ergebnisse in Lost Earth“ ist am niedrigsten eingestuft, da diese Anforderung erfahrungsgemäß einfach zu implementieren ist, wenn die Kommunikation zwischen LE und LRS steht.

7.2 Umsetzung der Anforderungen

In diesem Kapitel werden die einzelnen Anforderungen näher betrachtet und das Vorgehen der Umsetzung beschrieben. Um überhaupt zu testen, ob die Technologien mit Unity und den Tools zusammenspielen, wird ein Proof-of-Concept in Unity implementiert. Dafür werden die Anforderungen auf einer „grünen Wiese“, d.h. in einer leeren Entwicklungsumgebung, entwickelt. Gründe dafür sind, dass zum einen der Überblick des Projektes bewahrt wird und zum anderen stehen bei der Implementierung keine weitere Faktoren im Weg.

7.2.1 Proof-of-Concept

Um zu prüfen, ob die einzelnen Komponenten mit den Unity-Spiel zusammenspielen können, wird ein "Proof-of-Concept" erarbeitet. Hierbei wird eine minimalistische Szene in Unity erstellt. Diese Szene ist in der Abbildung 7.1 zu sehen. Lediglich nur ein Bildschirm mit einem Tisch und zwei Pflanzen reichen aus, um die Anforderungen in dieser Szene umzusetzen. Die 3D-Vorlagen vom Bildschirm und den Pflanzen werden kostenlos von der Webseite free3d.com angeboten.

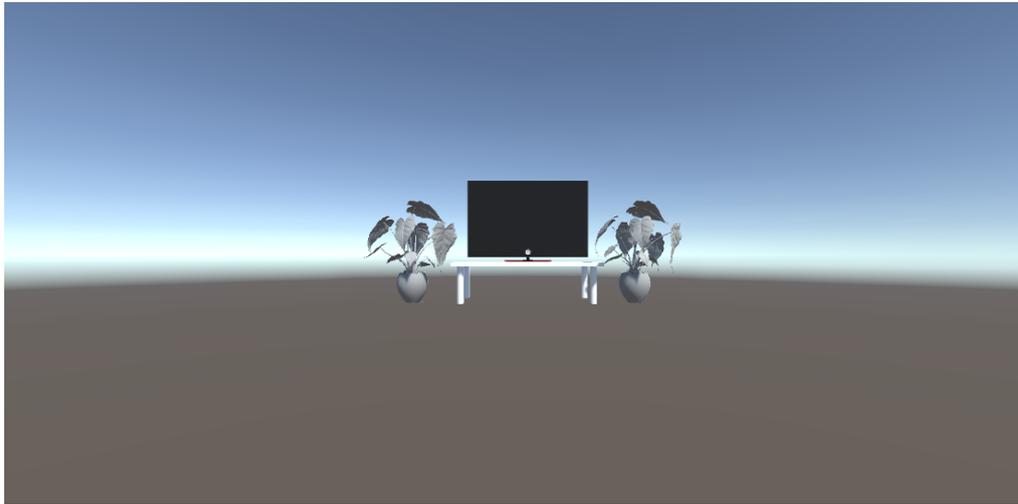


Abbildung 7.1: Dies zeigt die Proof-of-Concept-Umgebung. Es wird eine minimalistische Szene erstellt, in der die Kopplung der Komponente getestet wird.

7.2.2 Umsetzung 1: Anzeigen der externen Lerninhalte

Für das Anzeigen externen Lerninhalte wird ein eingebetteter Browser verwendet. Das Ziel ist es, dass man so gut wie alle Lern-Tools im Netz anzeigen möchte und dies wird nur erreicht, wenn das Spiel selbst einen Browser verwendet. Ein Browser in Unity kann auf verschiedene Weise umgesetzt werden. Früher gab es einen Browser als uWebKit für Unity, was leider mittlerweile nicht mehr verfügbar ist. Da die Unity Community sehr aktiv ist, gab es mehrere Unity Browser. Für die Arbeit wird der vielversprechende Simple Unity Browser von Vitaly Chashin verwendet. Dieser Browser basiert auf dem Chromium Embedded Framework und ist in der Lage im Unity-Spiel Webseiten aufzurufen und anzuzeigen, vor- und rückwärts zu gehen und mit Webseiten zu interagieren (siehe Abbildung 7.3). Dieser Browser kann zusätzlich Musik, Videos und webbasierte Spiele abrufen.

Für das Integrieren des Simple Unity Browsers musste nur das Projekt heruntergeladen, in Unity importiert und konfiguriert, beispielsweise muss man den lokalen Speicherort in einem Script vom Browser angeben, damit der Browser die mitgelieferten Serverplugins wieder finden.

Um auf den Bildschirm klicken zu können, dadurch die Sicht zum Bildschirm zu vergrößern und die initialisierte Seite im Browser anzuzeigen, wurde die von Unity bereitgestellten Werkzeuge verwendet. Die Werkzeuge sind C-Sharp-Scripte und bieten viele Mechanismen an, die alle in [Unic] aufgezeichnet sind. Für das Verständnis was Unity anbietet, wird im Folgenden ein kleiner Teil von Unity erläutert. Ein Beispiel, das detailliert betrachtet wird, ist das Anklicken

des Bildschirms, welcher daraufhin Funktionen ausführt.

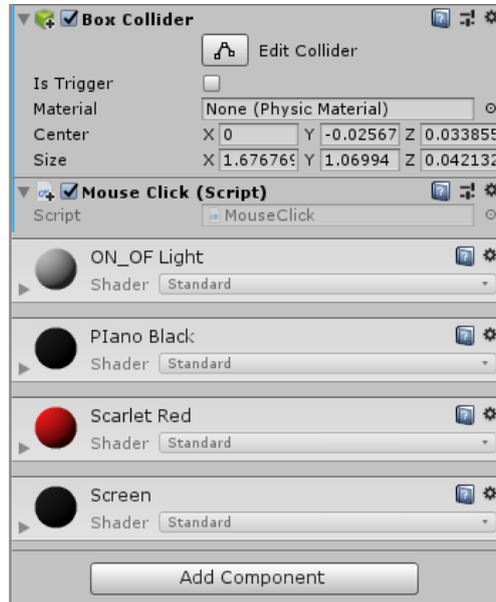


Abbildung 7.2: Ein Inspector Fenster ist zu sehen und zeigt ein GameObject mit ein Collider und benutzerdefiniertes Skript als Komponente

Um ein Objekt in der Szene anklicken zu können, muss das Objekt einen sogenannten „Collider“ als Komponente besitzen (siehe Abbildung 7.2.2). Dieser Collider ist dafür da, alle Formen von Kollisionen zu detektieren. In unserem Fall ist es ein unsichtbarer Strahl (im englischen „Ray“), der in die Richtung des Mauszeigers in der virtuellen 3D-Welt schießt. Der Strahl trifft in diesem Fall den Bildschirm und der Collider wird aktiviert. Um diesen Collider zu verwenden, reichen wenige Klicks aus, nämlich indem man auf den Plus-Button unten in der Abbildung 7.2.2 klickt und den Collider hinzufügt. Weiter unten ist eine Funktion zu sehen, die vom Collider aufgerufen wird, wenn der Bildschirm angeklickt wird. Diese Methode ruft eine Funktion (Zeile 32) auf, die eine HTML-Seite im Browser öffnet und in Zeile 33 eine weitere, die die Ansicht näher an den Bildschirm heran bringt.

```

1  void OnMouseDown() {
2
3      browser.launchHtml();
4      cs.zoomIn();
5
6  }
```

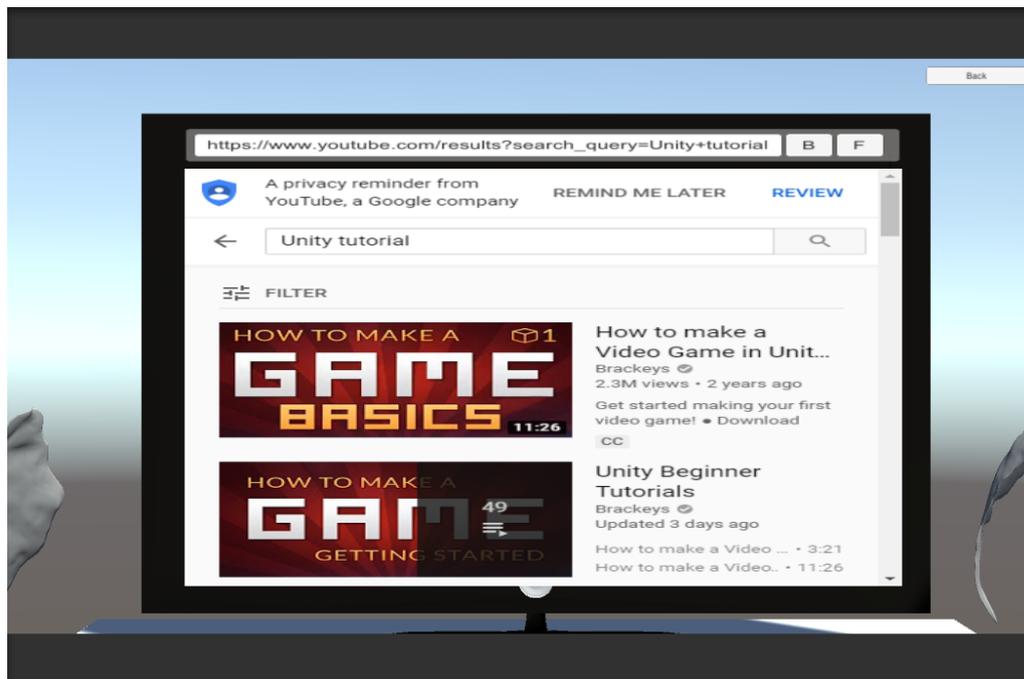


Abbildung 7.3: Ein Screenshot vom Unity-Prototyp, während man im Spiel nach Unity-Tutorials auf Youtube sucht.

7.2.3 Umsetzung 2: Integration des LTI-Lern-Tools

Für die Umsetzung der Integration eines LTI-Lern-Tools, muss das LE Spiel einen LTI-Aufruf mit den Parametern, die im Abschnitt 4.3.2 beschrieben sind, an das Tool geschickt werden. Dadurch wird auch die Authentifizierung umgesetzt (siehe Kapitel 4.3.2). Dafür wurde zuerst der eingebettete Browser näher untersucht. Der Browser besitzt eine JavaScript-Schnittstelle und kann lokale HTML-Dateien ausführen. Für das Aufrufen einer LTI-Anfrage wurde eine HTML-Datei verwendet. Diese Entscheidung ist gefallen, weil die Sprache fähig ist, JavaScript einzubinden und außerdem für das Anzeigen eigener Startseite nützlich ist.

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset ="UTF-8">
5 <title> Launch Website </title>
6 </head>
7 <body>
8     This is a LTI Lauche Page
9
```

```

10 <form id="ltiform"
    action="https://clm-moodle.iosb.fraunhofer.de/enrol/lti/tool.php?id=1"
    method="POST">
11
12 <input name="custom_component_display_name" type="hidden"
    value="CLM-LostEarth+LTI+Launcher+Provider">
13 <input name="lti_version" type="hidden" value="LTI-1p0">
14
15 <input name="oauth_nonce" type="hidden"
    value="61316703539413992421564044855">
16 <input name="resource_link_id" type="hidden"
    value="-66738e6fdafb4b35805e1bbc63af7f52">
17 <input name="context_id" type="hidden"
    value="course-v1:FraunhoferIOSB+clm1+2018-12">
18
19 <input name="context_title" type="hidden" value="ClmUnityLtiTestCourse">
20 <input name="lti_message_type" type="hidden"
    value="basic-lti-launch-request">
21 <input name="launch_presentation_return_url" type="hidden" value="">
22 <input name="context_label" type="hidden" value="FraunhoferIOSB">
23 <input name="user_id" type="hidden" value="student">
24 <input name="roles" type="hidden" value="Lerner">
25 <input name="oauth_consumer_key" type="hidden"
    value="moodle-anything-i-want-key">
26 <input name="lis_result_sourcedid" type="hidden"
    value="course-v1:-66738e6fd2:student">
27
28 <input name="oauth_signature_method" type="hidden" value="HMAC-SHA1">
29 <input name="oauth_version" type="hidden" value="1.0">
30
31 <!-- Changes this with actual timestamps!-->
32 <input name="oauth_signature" type="hidden"
    value="kXgMfEM09yCLzVVxli9PZpbQaac=">
33 <!-- Timestamp -->
34 <input name="oauth_timestamp" type="hidden" value="1564044855">
35 <input name="oauth_callback" type="hidden" value="about:blank">
36
37 <!--This is a submit Button-->
38 <input type="submit" value="Launch lti Moodle curse">
39 </form>
40 </body>
41 </html>

```

Diese Code-Zeilen stellen den ersten Ansatz einer HTML-Datei dar, die einen beispielhaften

LTI-Aufruf mit hart-codierten Parametern ausführt. Im Vorfeld muss erwähnt werden, dass dieser Aufruf nicht funktioniert, wenn die Oauth-Parameter invalide sind. In den Zeilen 1-7 und 40-41 sind der Standardaufbau eines HTML-Datei zu sehen. Kurz gesagt, die Datei wird mit `<>` und `< />` Notation in Bereiche unterteilt. Eine Form ist in HTML (hier Zeile 10-39) dafür da, um Eingaben zu sammeln [W3S]. In Zeile 9 wird definiert, welche Aktion beim Absenden der Form gemacht wird. Beim Absenden wird die Form an die CLM-Moodle-Plattform mit einer HTTP-Post-Methode gesendet. Wie in Kapitel 4.3.2 beschrieben, werden die Parameter in die Form als Inputs eingesetzt (Zeile 12-38). Die Inputs geben jeweils den Namen des Parameters an und den dazugehörigen Inhalt. Die Inputs sind als Typ „hidden“ festgelegt, um zu verhindern, dass der Benutzer die Parameter sehen oder modifizieren kann [MDN]. Wie schon erwähnt, funktioniert der Aufruf nicht, wenn die Oauth-Parameter nicht valide sind. Die Ursachen bei einer Invalidität sind verschieden. An dieser Stelle werden exemplarisch nur die Oauth-Parameter betrachtet. Nähere Beschreibungen sind im Kapitel 4.3.2 zu lesen. Für einen validen Aufruf wird zusätzlich JavaScript-Code in die HTML-Datei eingebunden.

```
1 function oauthsetup() {
2     var oauth = require('oauth-sign');
3     var action = 'https://clm-moodle.iosb.fraunhofer.de/enrol/lti/tool.php';
4     var method = 'POST';
5     var timestamp = Math.round(Date.now() / 1000);
6     var params = {...};
7     // to signature you have to put the secret here
8     var signature = oauth.hmacsign(method, action, params, 'secret');
9     params.oauth_signature = signature;
10    var form = document.querySelector("#ltiForm");
11    form.action = action;
12    form.method = method;
13    for (var name in params) {
14        var node = document.createElement("input");
15        node.name = name;
16        node.type = 'hidden';
17        node.value = params[name];
18        form.appendChild(node);
19    }
```

Wenn der Timestamp nicht mehr aktuell ist, wird der Aufruf nicht akzeptiert. Aus diesem Grund wird der aktuelle Timestamp mit der Methode „Date.now()“ geholt (Zeile 5). Die Methode gibt den Zeitstempel in Millisekunden zurück und muss in ganze Sekundeneinheiten umgewandelt werden. Der Aufruf kann ebenfalls invalide sein, wenn der Oauth-Nonce-Wert zweimal hintereinander verwendet wird. Dieser wird in der Regel zufällig generiert. Die Signa-

tur geht über alle Parameter, auch die Methode und Action von der Form werden signiert. Dafür wird zusätzlich noch ein JavaScript-Bündel von NPM eingebunden, das die Hashsignierung mitbringt (Zeile 7) [NPM].

Nachdem die Oauth-Parameter generiert wurden, kann die LTI-Anfrage geschickt werden und die Authentifizierung läuft im Hintergrund ab. Man erhält anschließend eine Antwort vom Tool und darin enthält der externe Lerninhalt, die im Browser angezeigt wird. Für den Anfang wurde mit LTI eine Webseite aufgerufen, die ein LTI-Tool simuliert. Diese Seite zeigt an, ob der LTI-Aufruf eine gültige Syntax hat und prüft zusätzlich die Korrektheit der Signierung [IMSc].

7.2.4 Umsetzung 3: Integration des LRS

Damit das LE von den Lernergebnissen oder Lernprozessen informiert wird und selbst Benutzerinteraktionen an beispielsweise andere Lernplattformen senden kann, wird ein Learning Record Tool integriert. Da GBLxAPI mit der Demonstration eines Spiels mit xAPI gezeigt haben (siehe Abschnitt 5.3), ist eine eigene Integration xAPI möglich.

Wie die Notation der xAPI-Statements lautet, wurde bereits in den Grundlagen aufgegriffen (siehe Kapitel 4.3.3). Kurz zusammengefasst, ein Statement besteht mindestens aus Subjekt, Verb und Objekt. Das Subjekt führt die Lernaktivität aus, das Verb beschreibt die Aktivität und das Objekt erläutert näher, was das Subjekt bei der Aktivität gemacht hat.

Für das Verwenden von xAPI in C-Sharp gibt es bereits die Bibliothek TinCan.NET [rus]. Diese Bibliothek kann nur ausgeführt werden, wenn die Umgebung .NET 3.5 oder höher unterstützt wird. Unity unterstützt viele .NET Profile, z.B. die Anforderung der Bibliothek [Unia]. Somit ist Unity in der Lage die TinCan.NET Bibliothek zu verwenden.

Im Folgenden wird die Nutzung der Bibliothek anhand eines Programmstück in Unity erklärt. Für die Schnittstelle eines LRS wird eine Klasse TinCanInterface erstellt (Zeile 9). In Zeile 1-5 sind Standardbibliotheken, die man für den Normalgebrauch eines Unity-Skriptes benötigt. In Zeile 6 und 7 wird die TinCan.NET-Bibliothek importiert. Für den Konstruktor der Klasse (Zeile 17) muss man den Endpoint, den Schlüssel und das Secret des LRS-Frameworks als Parameter angeben. Die Parameter werden auf der Instanz gespeichert (Zeile 18-20) und ein LRS-Objekt von TinCan initialisiert, indem man die drei Parameter ebenfalls in den Konstruktor weiter gibt (Zeile 21). Mit diesem LRS-Objekt kann man nun das externe LRS ansprechen und Statements aufrufen und speichern. Für den Gebrauch im LE werden die Funktionen SendStatement (Zeile 24) und getQueryStatement (Zeile 26) explizit angepasst.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using System;
4 using UnityEngine;
```

```
5 using UnityEngine.UI;
6 using TinCan;
7 using TinCan.LRSResponses;
8
9 public class TinCanInterface{
10
11     private string endpoint;
12     private string user;
13     private string pass;
14     // from TinCan a Learning Record Store
15     private RemoteLRS lrs = null;
16
17     public TinCanInterface(string endpoint, string user, string password) {
18         this.endpoint = endpoint;
19         this.user = user;
20         this.pass = password;
21         this.lrs = new RemoteLRS (endpoint, user, password);
22     }
23
24     public LRSResponse SendStatement(Statement statement) {...}
25
26     public List<Statement> getQueryStatement(StatementsQuery query) {...}
27 }
```

Nachdem die Implementierung der Schnittstelle des LRS umgesetzt wurde, konnte das LE diese Funktionen nutzen. Beispielsweise sendet das Serious Game beim Starten ein xAPI-Statement mit dem Inhalt „LernendeX startete LE“ an den LRS. Zusätzlich wurde die Funktion implementiert, dass das LE ein spezifisches Statement vom Lernenden abfragt, um heraus zu finden, ob der Lernende einen Kurs abgeschlossen hat oder nicht. Beim Abschluss wird darauf hinaus das Bewertungsergebnis aus den Statements gelesen.

7.2.5 Umsetzung 4: Übertragung der Benutzerdaten an das LE

Bei dieser Umsetzung müssen Benutzerdaten von der CLM-Plattform an den LE übertragen werden. Die Problematik ist, dass das LE und die Plattform vollständig voneinander abgeschottet sind. Aus diesem Grund wird eine Zwischenkomponente für die Benutzerdaten benötigt. Die Komponente ist ein LTI-Tool, weil es ein externes Werkzeug ist mit Funktion Benutzerdaten zu speichern. Für den LTI-Tool wurde der Tiny LTI Provider von Github [Tor] benutzt. Dieses ist ein vollständig lauffähiges LTI-Tool und musste auf einem Server aufgesetzt werden. Heroku bietet Plattformen als Dienste an. Eine Besonderheit ist, dass Heroku kostenlos die Dienste bereitstellt. Infolgedessen wurde das Tool auf Heroku aufgesetzt und ist nun für LE und CLM-

Plattform erreichbar. Die CLM-Plattform ruft das Benutzerdaten-Tool mit einem LTI-Aufruf auf. In diesem Aufruf befinden sich die Nutzerdaten und das Tool speichert diese Daten für eine gewisse Zeit zwischen. Wenn das LE eine Get-Anfrage an das Tool sendet, schickt es die zwischengespeicherten Benutzerdaten als Antwort zurück. Damit wurde die Übertragung der Benutzerdaten von der CLM-Plattform und LE umgesetzt.

Eine Schwierigkeit war es, dass die Komponente LE und der eingebetteter Browser keine Schnittstelle für den Datenaustausch hatten, obwohl der Browser in LE integriert ist. Gelöst wurde dieses Problem, indem das LE die Benutzerdaten vom LTI-Benutzerdaten-Tool anruft und eine JavaScript-Datei mit den Daten erstellt. Der Browser kann nun die Daten auslesen und besitzt ebenfalls die Benutzerdaten. Die Datei muss eine JavaScript-Datei sein, da es ansonsten Probleme mit dem CORS des Browsers gibt. Der Aufruf und das Erstellen der Datei wird wie im Folgenden umgesetzt.

```

1 public void Request()
2     {
3         StartCoroutine(GetRequest(ltiUserDataToolURL));
4     }

```

Für das Abfragen der Benutzerdaten soll der laufende Prozess nicht blockiert werden, da eine Abfrage mehrere Sekunden dauern kann. In Unity gibt es dafür die StartCoRoutine-Funktion [Unic]. Diese Funktion ruft eine andere Funktion auf, die es erlaubt die Ausführung jederzeit anzuhalten und dort weiter macht, wo es angehalten wurde. Beim Pausieren erledigt Unity wichtigere Funktionen und blockiert somit nicht den Prozess. In Zeile 3 ist so ein Aufruf zu sehen. Unity startet mit der Funktion eine Coroutine und führt eine eigene GetRequest()-Funktion mit der URL vom Benutzerdaten-Tool als Parameter aus.

```

1 IEnumerator GetRequest(string uri) {
2     using (UnityWebRequest webRequest = UnityWebRequest.Get(uri)) {
3         // Request and wait for the response
4         yield return webRequest.SendWebRequest();
5         // from Lti User Data Tool you will get an Json. Otherwise you will get
           a html data
6         string json = webRequest.downloadHandler.text;
7         // filter and save LTI params
8     try {
9         // Save MetaData in a json file
10        writeJsonDown(json);
11        // JObject parse the Json in a "Value Key List". Access the data
           value via data[objectName]
12        JObject data = JObject.Parse(json);
13        //set the user data

```

```
14         userdata = new UserData(  
15             data["lis_result_sourcedid"]["data"]["userid"].ToString(),  
16             data["ext_user_username"].ToString(),  
17             data["lis_person_contact_email_primary"].ToString(),  
18             data["lis_person_name_family"].ToString(),  
19             data["lis_person_name_full"].ToString(),  
20             data["lis_person_name_given"].ToString(),  
21             data["roles"].ToString());  
22     } catch(System.Exception e) {  
23         dataIsSet = false;  
24         Debug.LogError("Error: " + e);  
25     }  
26 }  
27 }
```

Die Coroutine führt die obere GetRequest-Funktion aus. Um diese anhalten und fortfahren zu können, wird ein Interface IEnumerator verwendet (Zeile 1). Dieser erlaubt die Codeausführung zu steuern. In diesem Fall wird sie angehalten und fortgeführt. Using ist eine Art Makro und substituiert webRequest mit UnityWebRequest.Get(uri) (Zeile 2). In Zeile 4 wird die Anfrage gesendet und gewartet bis eine Antwort empfangen wird. Nach der Anfrage befindet sich die Antwort im UnityWebRequest-Objekt und kann mit downloadHandler.text die JSON-Datei ausgelesen werden (Zeile 6). Die JSON-Datei wird mit einer eigenen Funktion als JavaScript-Datei gespeichert (Zeile 10). Da der JSON-String den ganzen LTI-Aufruf von der CLM-Plattform enthält, müssen die Benutzerdaten ausgefiltert werden (Zeile 12-22). Dazu wird der JSON-String in eine Art Value-Key-Liste geparsed (Zeile 12) und speichert die benutzerspezifischen Daten in einem UserData-Object (Zeile 15-22). Normalerweise kann man JSON-Strings nicht in ein Objekt parsen und daher wird die Newtonsoft-Framework verwendet [New]. Somit erhält das LE die Benutzerdaten und reicht sie an den Browser weiter, indem die HTML-Datei den JavaScript einbindet.

7.2.6 5. Umsetzung: Anbieten der externen Lernmaterialien für das LE

Für diese Umsetzung wurde die Plattform Moodle verwendet. Mit dieser Plattform kann man einfach und bequem einen LTI-Kurs erstellen, allerdings muss man dafür die notwendigen Rechte besitzen. Zuerst muss ein normaler Kurs angelegt werden, um anschließend den Kurs als LTI-Tool freizugeben. Dieses Vorgehen ist wegen der benutzerfreundlichen Moodle-Seite trivial und wird nicht näher erläutert. Bei der Freigabe generiert Moodle die URL zum Aufrufen, einen Key und ein Secret, mit dem man den LTI-Aufruf signiert. Die Zugangsdaten werden beim LTI-Aufruf eingesetzt und somit kann der Prototyp aufgerufen und angezeigt werden

(siehe Abbildung 7.4).

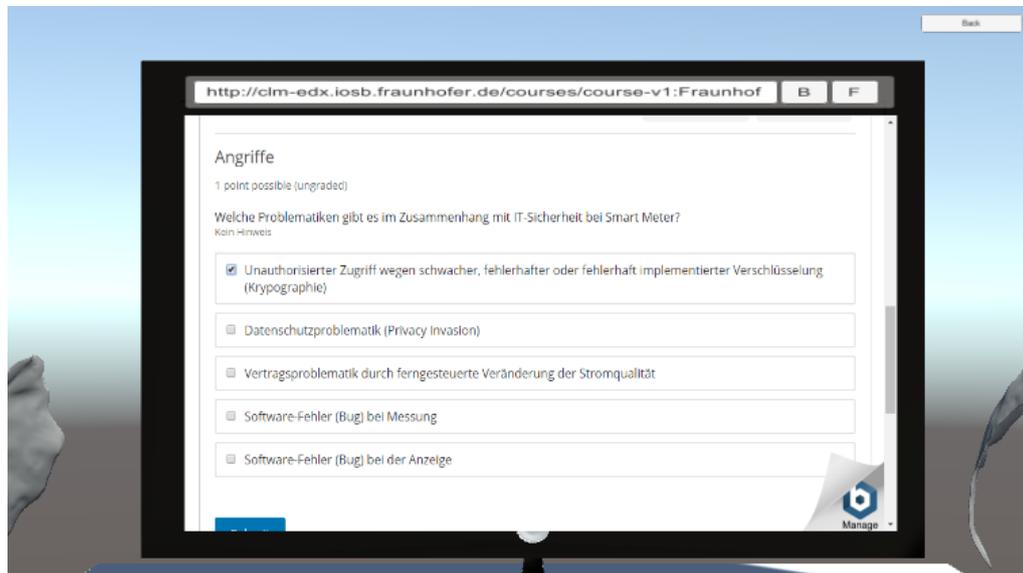


Abbildung 7.4: Der Prototyp ruft einen LTI-Tool auf und zeigt es auf dem Bildschirm an. Diese Seite wurde mit LTI aufgerufen.

7.2.7 Umsetzung 5: Einfluss der Ergebnisse in Lost Earth

Da die Anforderung 3 umgesetzt wurde, ruft der Prototyp das Ergebnis vom LRS auf und gibt ein Feedback an die Lernenden. Die eigentliche Umsetzung dieser Anforderung wird im folgenden Kapitel erläutert.

7.3 Umsetzung in LE

Der Prototyp zeigt, dass die Umsetzung der Anforderung möglich ist und somit kann er in das eigentliche Spiel integriert werden. Als erstes wurde die Ordnerstruktur vom gegebenen LE untersucht. In Unity gibt es einen Projektordner für Assets, worin alle Elemente vom Spiel sortiert sind (siehe Abbildung 7.5).



Abbildung 7.5: Gesamte Ordnerstruktur von LE

Die relevanten Ordner sind „Scripts“, in den die C-Sharp-Skripte gelagert sind, der „Plugins“-Ordner und der neu angelegter Ordner „Sandbox“. In diesem Ordner wurden vorerst alle Daten vom Proof-of-Concept hinzugefügt und anschließend getestet. Aufgrund von unterschiedlichen Versionen mussten die neuen Features leicht angepasst werden. Nachdem alle lauffähig waren, wurden die Dateien in die entsprechenden Ordner einsortiert (siehe dazu Abbildung 7.6). Der eingebettete Browser wurde in die Plugins eingeordnet und die Skripte für xAPI und LTI sind im Skript-Ordner zu finden.



Abbildung 7.6: In diesen Ordnern sind die Funktionen für xAPI, LTI und Benutzerdaten, sowie der eingebettete Browser zu finden

Da das LE mehrere Szenen besitzt, die eine Art Level, in dem man hin und her navigieren kann, und die ganze Umgebung und den Funktion darstellt, müssen über diese Szenen Daten ausgetauscht werden. Beispielsweise müssen die Benutzerdaten konstant verfügbar gemacht werden, sowie ein xAPI-Manager, um die Statements in jeder Szene empfangen oder versendet werden kann. Dafür wird die von Unity angebotene Methode „DontDestroyOnLoad()“ verwendet. Um diese Methode zu nutzen, wird in der ersten Szene „Lobby“ ein Unity-Objekt mit dem Skript erstellt. Das Skript ruft automatisch die Funktion „Awake()“ auf, wenn das Objekt mit dem Skript in der Szene erstellt wird (Zeile 1). Mit dem Aufrufen von DontDestroyOnLoad(transform.gameObject) wird das Objekt nicht mehr zerstört, wenn eine neue Szene geladen wird. Somit können die Skripte für Benutzerdaten und xAPI in jeder Szene verwendet werden.

```

1 private void Awake()
2     {
3         // this Object will not destroy by switching scenes. Unity destroy All
4         //   GameObject when loading a new level.
5         DontDestroyOnLoad(transform.gameObject);
6     }

```

Die gesamten Beziehungen des LE werden in der Abbildung 7.7 veranschaulicht. Wie in der Architektur in der Konzeption 6.4 beschrieben, holt sich das LE die Nutzerdaten vom User-Data-Tool, das Lernmaterial vom Lern-Tool und speichert und ruft xAPI-Statements vom

LRS ab. Im Inneren vom LE werden die Nutzerdaten vom UserDataManager abgerufen und verwaltet. Dieser stellt die Daten für den LTI-Aufruf und für den xAPI-Manager bereit. Der xAPI-Manager speichert und lädt die aktuellen relevanten Statements. Der Manager ruft den UserDataManager mit den Benutzerdaten für das Speichern und das Abfragen der Statements des Benutzers auf. Die LTI-HTML-Datei erhält die Benutzerdaten von der vom UserDataManager erstellte Datei. Der FeedbackController nimmt die Dienste vom xAPIManager in Anspruch und fragt nach aktuellen Statements. Beispielsweise, wenn ein Kurs abgeschlossen wurde, schickt der FeedbackController eine Nachricht an den Benutzer und teilt ihm mit, wie seine Ergebnisse waren und wie sich das auf das Spiel auswirkt. Der Spieler erhält zum Beispiel 1000 Wissenschaftswährungen.

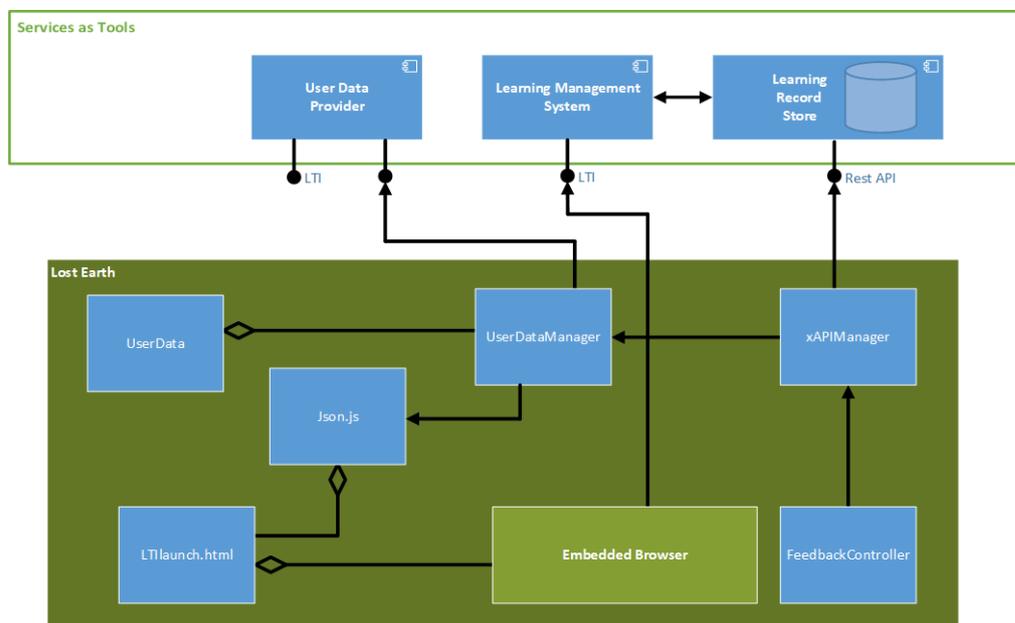


Abbildung 7.7: In dieser Abbildung sind die Beziehungen der Dateien ersichtlich. UserDataManager ruft vom User Data Provider die Benutzerdaten ab und gibt diese Daten weiter. Der xAPIManager ist dafür zuständig, xAPI-Statements zu speichern oder abzurufen.

Erwähnenswert ist hier noch, dass, wenn sich der Pfad von den Browser-Plugins im Assets-Ordner ändern sollte, die Pfade in „Scripts/BrowserEngine.cs“ und „Editor/BrowserPost-Build.cs“ aktualisieren werden, damit der Browser die Plugins wieder finden kann. Außerdem, wenn man den Speicherort der JavaScript-Dokument für die Benutzerdaten ändern möchte, muss man beim UserDataManager-Objekt in der UserMetaDataManager-Script-Komponente den neuen lokalen Pfad angeben 7.8. Zusätzlich muss im HTML-Dokument bei LTI der Speicherort angepasst werden, um die Daten lesen zu können.

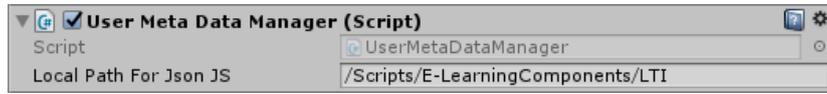


Abbildung 7.8: Konfiguration des Speicherorts für die JSON-Datei

8 Anwendungsszenario

In diesem Kapitel wird der Ansatz der Konzeption 6.3 aufgegriffen und anhand der umgesetzten Lernumgebung aus der Benutzersicht vorgeführt. Der Lernende meldet sich wie gewohnt an und startet das LE. Die Lobby ist die Startseite vom LE (Abbildung 8.1). In dieser Szene kann man unten rechts das Tracken der Benutzeraktivitäten ein- oder ausschalten. Mithilfe des Buttons in der Mitte startet man das Spiel.



Abbildung 8.1: Die se Abbildung zeigt die Lobby von LE. In der Mitte ist der Startbutton zu sehen und unten rechts kann das Tracken der Benutzeraktivitäten ein- oder ausgestellt werden.

Nachdem man den Startknopf gedrückt hat, wird das gesamte Schiff (Arche) mit den einzelnen Abteilungen angezeigt 8.2. Um zu den externen Lernmaterialien zu gelangen, muss auf die Brücke (oberste rechte Abteilung) geklickt werden.



Abbildung 8.2: Ansicht von der Arche

Bei der Brücke kann man die Konsole, die hier mit einem roten Pfeil markiert ist, anklicken, um die externen Lerninhalte aufzurufen 8.3.



Abbildung 8.3: Szene von der Brücke

Nun gibt es zwei Fälle. Im ersten Fall, dass der Lernende schon den Kurs bzw. den Test vorher schon komplett abgeschlossen hat, wird die Benachrichtigung, die in Abbildung 8.4 zu sehen ist, angezeigt.



Abbildung 8.4: Hier wird eine Benachrichtigung gezeigt, wenn der Kurs schon abgeschlossen wurde.

Im anderen Fall, wenn der Lernende den Kurs bzw. den Test nicht abgeschlossen hat, öffnet sich der Browser mit den externen Lerninhalten 8.5. Die Besonderheit hier ist, dass dieser Lerninhalt nicht öffentlich zugänglich ist und das LE dafür einen LTI-Aufruf mit der notwendigen Authentifizierungsdaten senden musste. Somit kann die Seite sicher vom Lern-Tool abgerufen werden.

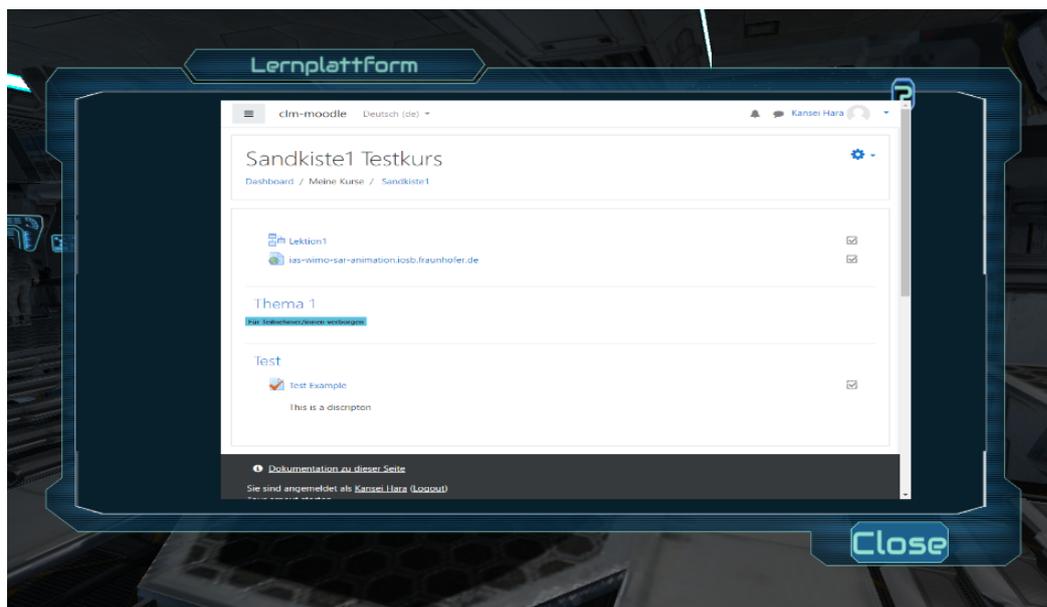


Abbildung 8.5: Ansicht der externen Lernmaterialien

Der Lernende kann, ohne den Kurs abzuschließen, die Anwendung verlassen, indem er auf den Schließ-Button klickt. In diesem Fall wird er aufgefordert, wieder zurückzukommen und den Kurs zu beenden 8.6.



Abbildung 8.6: Benachrichtigung bei Nicht-Vollenden des Kurses

Der andere Fall wäre, dass der Lernende den Kurs abschließt und anschließend den Browser schließt. Danach erhält der Lernende ein Feedback 8.7 zum Ergebnis und erhält je nach Ergebnis Wissenschaftspunkte. Zuletzt beendet der Benutzer das Lernen und schließt das Spiel und RPD.



Abbildung 8.7: Feedback beim Abschluss eines Kurses

9 Fazit und Ausblick

In diesem Kapitel wird zuerst das Ergebnis der Kopplung erläutert und diskutiert und zusätzlich wird ein Ausblick über die Fortführung der Entwicklung des Systems vorgeschlagen. Am Ende wird ein Fazit gezogen.

9.1 Ergebnis

In diesem Abschnitt wird das Ergebnis der Umsetzung der Kopplung vorgestellt. Mit der Kopplung von unterschiedlichen Lernumgebungen sind verschiedene Kommunikationswege entstanden. Mit dem Benutzerdaten-Tool ist das LE im RDP nicht mehr abgeschottet von der CLM-Plattform und ist damit möglich Benutzerdaten auszutauschen

Mit dem UserDataManager im LE ist es möglich, die Benutzerdaten an den ebenfalls abgeschotteten Browser weiter zu reichen. Die Kommunikation von Benutzeraktivitäten mit anderen Lernumgebungen wird mit dem LRS umgesetzt und das LE kann mit anderen Umgebungen die Ergebnisse oder die Lernaktivitäten vom Lernenden austauschen. Mit den reibungslosen Aufrufen von externen Lerninhalten im Spiel wurde eine nach dem momentanen Stand der Forschung noch nicht entwickelten Kopplung realisiert. Damit ist das LE fähig, wie ein Lernmanagementsystem zu agieren und flexibel externe Tools zu nutzen. Durch das Nutzen eines Browsers kann das LE jedes Lern-Tool aufrufen und anzeigen, was einen enormen Mehrwert darbietet. Der Mehrwert besteht darin, dass das Anzeigen der Lerninhalte komplett vom Browser übernommen wird und dass keine eigene Benutzerschnittstelle für jedes Lern-Tool implementiert werden muss.

9.2 Diskussion und Ausblick

Von einem anderen Blickwinkel betrachtet, entkoppelt das Benutzerdaten-Tool die Verwaltung der Sitzung vom Lernmanagementsystem an ein externes Tool. Man könnte daher dieses Tool als „Learning Session Tool“ bezeichnen, das für die Identifizierung des Nutzers verantwortlich ist. Momentan kann nur ein Benutzer das System verwenden, da das Tool nur eine Sitzung

verwaltet. Um mehrere Nutzer gleichzeitig das Spiel anbieten zu können, müssen mindestens genauso viele externe Computer verfügbar sein. Jeder Computer hat eine eindeutige ID. Das LE-Launcher-Tool kennt alle IDs und kann sie zu den jeweiligen RDP-Zugangsdaten zuordnen. Beim Belegen eines Computers sendet das Launcher-Tool eine Nachricht an das Learning-Session-Tool. Die Nachricht enthält die Nutzerdaten und bekommt als Antwort die RDP-Zugangsdaten. Das Session-Tool speichert währenddessen die Sitzung in einer Zeile mit Computer-ID und Nutzerdaten als Spalte in einer Tabelle. Das LE fragt das Session-Tool nach den Benutzerdaten und schickt dafür die ID mit. Das Session-Tool sucht die Zeile mit der ID und schickt die dazugehörige Benutzerdaten an das LE zurück.

Der Ansatz der Remote-Desktop-Applikation ist bereits vom Fraunhofer IOSB umgesetzt worden. Der Einsatz von RDP ist ein schneller und einfach zu realisierender Weg, um das Spiel allen Windows-Nutzern anzubieten. Diesen Ansatz kann man in ferner Zukunft erweitern. Das Cloud Gaming wäre eine moderne Alternative. Cloud Gaming bietet Spiele an, die auf einem Server laufen und Benutzerinteraktion über das Internet empfangen [Har10]. Die Kopplung des Lern-Tools erfolgt mit den LTI 1.1 Standard. Da LTI 1.3 sehr neu ist und noch nicht vollständig unterstützt wird, wurde dieser Standard verwendet. Die Sicherheit sind aufwendiger als der vorherige Standard, dafür aber um einiges sicherer. In der Zukunft könnte dies erweitert werden.

9.3 Fazit

Mit dieser gesamten Umgebung des E-Learnings wurde gezeigt, dass Lernplattformen nicht immer unflexible Lernspiele anbieten muss. Mit dieser Kopplung können Lernspiele beliebig viele Lerninhalte anbieten, die von externen Tools angeboten werden. Darüber hinaus können Lernaktivitäten mithilfe des LRS an andere Lernumgebungen geteilt werden. Mit dem Anbieten des LE-Spiels von der Lernplattform wird das Lernangebot mit einer motivierenden E-Learning-Umgebung und das Lernspiel um weit gefächerte Lerninhalte erweitert.

Literatur

- [Alfo6] S. S. Al-fedaghi. „Personal Management of Private Information“. In: *2006 Innovations in Information Technology*. 2006, S. 1–5.
- [Bak+17] Abdellah Bakhoui u. a. „Evolution of standardization and interoperability on E-learning systems: An overview“. In: *2017 16th International Conference on Information Technology Based Higher Education and Training, ITHET 2017* (2017), Chapter 14.
- [Bet09] Ellen Wagner Beth Davis, Colleen Carmen. „The evolution of the LMS: From management to learning“. In: *Santa Rosa, CA: e-Learning Guild* (2009). URL: <https://www.blackboard.com/resources/proed/guild-lmsreport.pdf>.
- [Bow13] Megan Bowe. *Its Time for Profiles*. 2013. URL: <https://xapi.com/blog/its-time-for-profiles/> (besucht am 07. 09. 2019).
- [Clai8] Stuart Claggett. *Getting Started with a Unity 3D xAPI Pilot*. 2018. URL: <https://gblxapi.org/community-blog-xapi-gbl/14-unity-3d-xapi-pilot-getting-started> (besucht am 12. 08. 2019).
- [Con] IMS Global Learning Consortium. *IMS Global Learning Consortium*. URL: <https://www.imsglobal.org/specifications.html> (besucht am 12. 08. 2019).
- [CRC02] By Geoff Collier, Robby Robson und Eduworks Corporation. „E-Learning Interoperability Standards“. In: *Learning Technology SUN microsystem* (2002).
- [Dag+07] Declan Dagger u. a. „Service-Oriented e-learning platforms: From monolithic systems to flexible services“. In: *IEEE Internet Computing* (2007).
- [DAJ11] Damien Djaouti, Julian Alvarez und Jean-Pierre Jessel. „Classifying serious games: The G/P/S model“. In: *Handbook of research on improving learning and motivation through educational games: Multidisciplinary approaches* (2011).
- [Dav13] Michelle R. Davis. *Big Tree*. 2013. URL: <https://www.edweek.org/dd/articles/2013/02/06/02textbooks.h06.html?cmp=ENL-DD-MOSTPOP> (besucht am 12. 08. 2019).

- [Del+13] Ángel Del Blanco u. a. „Using e-learning standards in educational video games“. In: *Computer Standards and Interfaces* 36.1 (2013), S. 178–187.
- [Eva+15] Kambria H. Evans u. a. „Septris: A novel, mobile, online, simulation game that improves sepsis recognition and management“. In: *Academic Medicine* (2015).
- [FMK11] Aroutis Foster, Punya Mishra und Matthew J Koehler. „Digital game analysis: Using the technological pedagogical and content knowledge framework to determine the affordances of a game for learning“. In: *Learning to play: exploring the future of education with video games*. 2011.
- [GH16] Dai Griffiths und Tore Hoel. „Comparing xAPI and Caliper Learning Analytics Review 7“. In: (2016). URL: http://www.laceproject.eu/learning-analytics-review/files/2016/08/LACE-review07%7B%5C_%7Dxapi-caliper.pdf.
- [Har10] Roland Austinat Hartmut Gieselmann Philipp Fechteler. „Über den Wolken. Wie Cloud Gaming den Spielmarkt revolutioniert“. In: *c't* 21 (2010), S. 76–83.
- [Her+10] Hal Herzog u. a. *IMS Learning Tools Interoperability Basic LTI Implementation Guide*. 2010. URL: <https://www.imsglobal.org/specs/ltiv1p0/implementation-guide> (besucht am 03. 09. 2019).
- [IEE07] IEEE Learning Technology Standards Committee. *IEEE Learning Technology Standards*. 2007. URL: <https://www.ieeeltsc.org/> (besucht am 12. 08. 2019).
- [IFF13] Borja Manero Iglesias, Clara Fernández-Vara und Baltasar Fernández-Manjón. „E-learning takes the stage: From La Dama Boba to a serious game“. In: *Revista Iberoamericana de Tecnologías del Aprendizaje* (2013).
- [IMSa] IMS Global Learning Consortium. *Caliper Analytics*. URL: <http://www.imsglobal.org/activity/caliper> (besucht am 12. 08. 2019).
- [IMSB] IMS Global Learning Consortium. *IMS Learner Information Package Specification*. URL: <https://www.imsglobal.org/profiles/index.html> (besucht am 12. 08. 2019).
- [IMSc] IMS Global Learning Consortium. *LTI-Validator*. URL: <http://dr-chuck.com/ims/php-simple/> (besucht am 12. 08. 2019).
- [IMS15] IMS Global Learning Consortium. *IMS Question & Test Interoperability (QTI) Specification*. 2015. URL: <https://www.imsglobal.org/question/index.html> (besucht am 01. 09. 2019).

-
- [IMS19] IMS Global Learning Consortium. *Learning Tools Interoperability Core Specification*. 2019. URL: <https://www.imsglobal.org/spec/lti/v1p3/> (besucht am 01.08.2019).
- [Ind19] ELearning Industry. *LTI Compliant LMS*. 2019. URL: <https://elearningindustry.com/directory/software-categories/learning-management-systems> (besucht am 09.08.2019).
- [Kevo3] Mark McKell (IMS) Kevin Riley (IMS). *IMS Digital Repositories Interoperability - Core Functions Information Model*. 2003. URL: https://www.imsglobal.org/digitalrepositories/driv1p0/imsdri%7B%5C_%7Dinfov1p0.html (besucht am 12.08.2019).
- [KH08] Mansureh Kebritchi und Atsusi Hirumi. „Examining the pedagogical foundations of modern educational computer games“. In: *Computers and Education* (2008).
- [Lou] Louie Andre. *20 Best LMS Software Solutions of 2019 - Financesonline*. URL: <https://financesonline.com/top-20-lms-software-solutions/>.
- [LQ11] José Paulo Leal und Ricardo Queirós. „Using the Learning Tools Interoperability Framework for LMS Integration in Service Oriented Architectures“. In: *TECH-EDUCATION'11*. 2011.
- [MDN] MDN Web Docs. `<input type="hidden">`. URL: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/hidden>.
- [Mih18] Andy Miller. *Launching an LTI 1.3 Resource Link using OpenID Connect Third Party Login*. 2018. URL: <https://andyfmiller.com/2018/12/28/launching-an-lti-1-3-resource-link-using-openid-connect-third-party-login/> (besucht am 12.08.2019).
- [MJ09] Mayo M.J. *Video games: A route to large-scale STEM education?* 2009.
- [Moo19] Moodle.net. *Moodle Statistics*. 2019. URL: <https://moodle.net/stats/> (besucht am 20.08.2019).
- [New] Newtonsoft. *Json.NET*. URL: <https://www.newtonsoft.com/json>.
- [NPM] NPM. *Hmac-Sign*. URL: <https://www.npmjs.com/package/hmac-sign>.
- [Pop+11] Maria Popescu u. a. „Serious games in formal education: Discussing some critical aspects“. In: *Proceedings of the European Conference on Games-based Learning*. 2011.
- [Row94] Derek Rowntree. *Preparing Materials for Open, Distance and Flexible Learning. An action guide for teachers and trainers*. 1994.

- [rus] rusticisoftware. *TinCan.NET*. URL: <http://rusticisoftware.github.io/TinCan.NET/> (besucht am 01. 09. 2019).
- [Sim18] Ormond Simpson. *Supporting students in online, open and distance learning*. 2018.
- [Sof] Rustici Software. *xAPI Overview*. URL: <https://xapi.com/overview/> (besucht am 04. 08. 2019).
- [Squ+04] Kurt Squire u. a. „Electromagnetism supercharged!:learning physics with digital simulation games“. In: *Proceedings of the 6th international conference on Learning sciences*. 2004.
- [The] The Dublin Core Metadata Initiative. *Metadata Basics*. URL: <https://www.dublincore.org/resources/metadata-basics/> (besucht am 01. 09. 2019).
- [Theo4] The Advanced Distributed Learning Initiative. *SCORM Overview*. 2004. URL: <https://adlnet.gov/research/scorm/> (besucht am 01. 09. 2019).
- [Tor] Luis de la Torre. *Python LTI Provider*. URL: https://github.com/UNEDLabs/tiny%7B%5C_%7Dlti%7B%5C_%7Dprovider (besucht am 24. 08. 2019).
- [TV97] Eilif Trondsen und Kent Vickery. „Learning on Demand“. In: *Journal of Knowledge Management* (1997).
- [Tyl16] Tyler Reed. *McGraw-Hill and LTI*. 2016. URL: <https://www.mheducation.com/news-media/press-releases/connect-integration-canvas-learning-management-system.html> (besucht am 15. 08. 2019).
- [Unia] Unity. *.NET Profile Support*. URL: <https://docs.unity3d.com/2018.3/Documentation/Manual/dotnetProfileSupport.html> (besucht am 01. 09. 2019).
- [Unib] Unity. *Unity 2019: Performance by default, high-fidelity real-time graphics, and artist tools*. URL: <https://unity3d.com/unity> (besucht am 04. 08. 2019).
- [Unic] Unity. *Unity Documentation*. URL: <https://docs.unity3d.com/ScriptReference/> (besucht am 12. 08. 2019).
- [VW03] G Vossen und P Westerkamp. „E-learning as a Web service“. In: *Sevent International Database Engineering and Applications Symposium (IDEAS'03)*. 2003.
- [W3C] W3C Community Development. *EXPERIENCE API (XAPI) VOCABULARY & SEMANTIC INTEROPERABILITY COMMUNITY GROUP*. URL: <https://www.w3.org/community/xapivocabulary/> (besucht am 24. 08. 2019).

- [W₃S] W₃Schools. *HTML Forms*. URL: https://www.w3schools.com/html/html%7B%5C_%7Dforms.asp (besucht am 12. 08. 2019).
- [Weg96] Peter Wegner. „Interoperability“. In: *ACM computing surveys*. Citeseer. 1996.
- [WGo9] Michael Wunder und Jürgen Grosche. „Überblick über Interoperabilitätsstandards“. In: *Verteilte Führungsinformationssysteme*. 2009. URL: https://link.springer.com/chapter/10.1007%7B%5C%7D2F978-3-642-00509-1%7B%5C_%7D14.

Abbildungsverzeichnis

4.1	Einteilung der etablierten Interoperabilitätsstandards	9
4.2	Allgemeines Komponentendiagramm eines Netzwerksystems mit LTI	12
4.3	Sequenzdiagramm eines LTI 1.3 Aufrufes	15
4.4	Die Unity Game Engine Entwicklungsumgebung	17
5.1	Integrationsarten eines Lernspiels	20
6.1	Aktivitätsdiagramm der allgemeinen Anforderungen	22
6.2	Sequenzdiagramm: Das Lernspiel Lost Earth starten	27
6.3	Sequenzdiagramm: Externe Lerninhalte im Lost Earth aufrufen	28
6.4	Sequenzdiagramm: Benutzeraktivitäten vom Lost Earth speichern	29
6.5	Sequenzdiagramm: Aufgabe oder Test bewerten	30
6.6	Sequenzdiagramm: Start- und Beenden-Statement speichern	31
6.7	Die Komplette Architektur der Kopplung der Lernsysteme	32
7.1	Proof-of-Concept-Umgebung	35
7.2	Unity-Inspector-Fenster	36
7.3	Anzeigen einer Webseite im Spiel	37
7.4	Prototyp: LTI-Lerninhalt anzeigen	44
7.5	Gesamte Ordnerstruktur von LE	45
7.6	Speicherort von xAPI, LTI und Browser im LE	46
7.7	LE-UML-Diagramm	47
7.8	Konfiguration des Speicherorts für die JSON-Datei	48
8.1	LE: Lobby	49
8.2	LE: Arche	50
8.3	LE: Brücke	51
8.4	LE: Benachrichtigung: Kurs vorher schon erledigt	52
8.5	LE: Ansicht der externen Lernmaterialien	53
8.6	LE: Benachrichtigung bei Nicht-Vollenden des Kurses	54

8.7	LE: Feedback beim Abschluss eines Kurses	55
-----	--	----