

QTI meets 3D – e-Assessment im dreidimensionalen Raum

Andreas Stahl, Stefan Gumhold

Professur für Computergraphik und Visualisierung
Technische Universität Dresden
Nöthnitzer Straße 46
01154 Dresden

andreas.stahl@tu-dresden.de
stefan.gumhold@tu-dresden.de

Abstract: Mit der Unterstützung von Hardware-beschleunigter 3D-Graphik in modernen Web-Browsern ist die Integration von dreidimensionalen Inhalten auf Webseiten möglich geworden. Am Beispiel des QTI-Standards wird gezeigt, wie 3D-Inhalte die Lernerfolgskontrolle in e-Learning-Angeboten bereichern können, was insbesondere im Bereich der Ingenieurs- und Lebenswissenschaften einen unschätzbaren Mehrwert bietet. Dazu wird ein Satz von 3D-Aufgabentypen für das e-Assessment vorgeschlagen und dieser prototypisch als Erweiterung von QTI implementiert.

1 Einleitung

In vielen Fachgebieten werden Lernende mit Lernobjekten im dreidimensionalen Raum konfrontiert, sei es in den Lebenswissenschaften oder in konstruktiven Ingenieursdisziplinen. Dabei ist die Demonstration von Sachverhalten sowie die Wissensüberprüfung am realen Objekt oftmals nicht praktikabel. So ist der Zugriff durch ungeübte Lernende oft riskant, einerseits für das Lernobjekt – etwa bei biologischen Organismen, andererseits aber auch für die Lernenden im Falle von Gefahrgut. Weiterhin spielen sich viele essentielle Vorgänge im Inneren eines Objekts, in extrem kurzen Zeitspannen, oder in sub-mikroskopischen Bereichen ab, so dass sie im Virtuellen erheblich leichter erfahrbar gemacht werden können als in der Realität. Die Verwendung von Virtual-Reality- und Web3D-Techniken im Dienste des e-Learnings ist somit ein zunehmend wichtiger Forschungsgegenstand (vgl. [MMB08] und [CR07]).

Im Zuge der Zurverfügungstellung von Lernmaterialien im World-Wide-Web steht – noch vor Aspekten der Datenbeschaffenheit oder der Standardisierung von Datenaustauschformaten – die Frage, welche Interaktionsmöglichkeiten im Dreidimensionalen überhaupt sinnvoll für die Lernerfolgskontrolle genutzt werden können. Im Folgenden werden daher verschiedene Aufgabentypen erarbeitet, die die Möglichkeiten dreidimensionaler Web-basierter Interaktionen nutzen. Die Definitionen dieser 3D-Aufgabentypen werden ausgehend von gebräuchlichen Aufgabentypen wie zum Beispiel Multiple-Choice, oder Lückentext vorgenommen.

Um virtuelle dreidimensionale Interaktionen nicht nur für die Wissensvermittlung, sondern auch die Lernerfolgskontrolle nutzbar zu machen, ist eine Vielzahl von Problemen zu lösen, die die Erzeugung, Verarbeitung und Präsentation der Lernobjekte betreffen. Die Lernobjekte müssen in einem 3D-Autorenprozess beispielsweise mit Markierungen versehen werden, die auf Auswahl durch die Lernenden reagieren. Diese so aufbereiteten Daten müssen dann von gebräuchlichen Lernmanagement-Systemen wie Moodle, Blackboard oder OLAT verarbeitet werden können, was voraussetzt, dass sie in einem standardisierten Datenformat vorliegen.

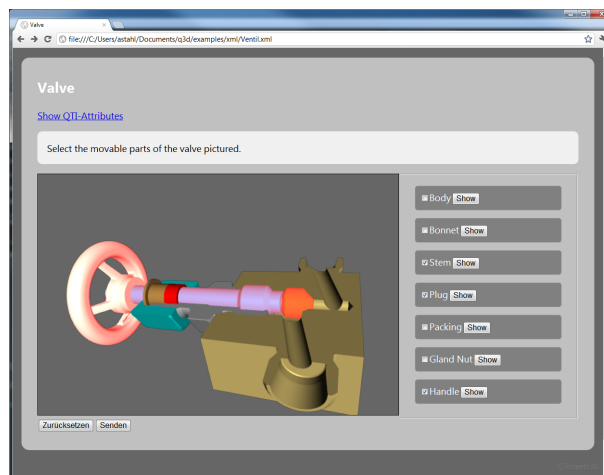


Abbildung 1: Ein Bildschirmfoto des Prototypen. Dargestellt ist eine Multiple-Picking-Aufgabe aus dem Bereich Maschinenbau. Lernende sollen die beweglichen Teile eines Ventils markieren. Dies kann über die Abbildung direkt oder indirekt über die Formular-Elemente an der rechten Seite erfolgen.

In diesem Beitrag wird eine Schema-Erweiterung des weit verbreiteten Datenaustauschformats IMS Question and Test Interoperability Version 2.1 PD2 (QTI [IMS06]) vorgestellt, sowie eine Referenzimplementierung der Client-Funktionalität mittels XML-Stylesheets (XSL-T), HTML5 und Javascript skizziert. Abbildung 1 zeigt ein Bildschirmfoto dieses Prototypen. Die im Folgenden vorgestellte Erweiterung ermöglicht es, dreidimensionale Objekte und Daten interaktiv in e-Assessment-Situationen darzustellen. Unterstützt werden Daten im Extensible-3D-Format (X3D) des Web3D-Konsortiums ([Web08]), die mittels der X3DOM-Javascript-Bibliothek ([BEJZ09]) dargestellt werden.

2 Verwandte Arbeiten

Die grundlegenden Charakteristiken des Einsatzes von dreidimensionalen, virtuellen Lernumgebungen werden von Chittaro und Ranon ([CR07]) dargelegt: interaktive dreidimensionale Modelle können Lernenden Sachverhalte, Vorgänge und Objekte offenlegen, deren

reales Erleben aus Gründen der Kosten, Gefahren, Entfernung oder zeitlichen Umständen, etwa bei Gebäuden der Antike, unmöglich ist. Die Autoren nennen als weitere Vorteile, dass die freie Wahl des Blickpunktes die Lernenden bei der explorativen Bildung mentaler Modelle unterstützt, und dass die Möglichkeit existiert, beim Distanzlernen *personae* der Mitlernenden und der Lehrpersonen in der Lernumgebung einzubinden. Als Nachteile werden Navigationsprobleme für ungeübte Nutzer sowie die hohe Erwartung hinsichtlich der Darstellungsqualität angebracht, die bei heutigen Umsetzungen oft enttäuscht wird, was sich negativ auf den Lernvorgang auswirkt.

Dass die interaktive dreidimensionale Präsentation von Lernmaterialien, zum Beispiel geometrischer Sachverhalte, die Lernenden beim Aufbau eines mentalen Modells unterstützen kann, wird auch von Yeh und Nason [YN04] belegt. Die Auswirkung einer interaktiven dreidimensionalen Darstellung auf die kognitive Belastung (Cognitive Load) hingegen ist Gegenstand aktueller Diskussionen (vgl. [Hö10]). Huk argumentiert, dass darauf geachtet werden müsse, dass die Darstellungsweise jene Lernenden mit schwachem räumlichen Vorstellungsvermögen nicht überfordert (vgl. [Huk06, Huk07]). Hier bietet die Verwendung dreidimensionaler Daten im Kontext des e-Assessments die Chance, indem dadurch das mentale Modell eines Lernenden, insbesondere seine räumliche Kenntnis eines Lerngegenstandes überprüfbar wird, wertvolle Beiträge in diese Diskussion einzubringen, und 3D-Lernmaterial effektiv auf seine Eignung zur Wissensvermittlung hin zu überprüfen.

Besonders in Lehrgebieten der Lebenswissenschaften und der Medizin ist die Arbeit mit dreidimensionalen Simulationen in der Regel einfacher und anschaulicher, als am realen Objekt. So existieren Untersuchungen zur Wissensvermittlung bezüglich medizinischer Verfahren [SPC⁺09], sowie die Einübung von Arbeitsabläufen medizinischer Geräte mit Unterstützung von dreidimensionaler Darstellungen innerhalb von Moodle [Çe12]. In beiden Untersuchungen konnte gezeigt werden, dass die Interaktivität dreidimensionaler Lernmaterialien Lernenden dabei hilft, komplexe Zusammenhänge zu verstehen und ihr Wissen dauerhaft zu festigen.

Die automatische Erzeugung von dreidimensionalen Daten aus einem anderen wissenschaftlichen Datenformat zur Beschreibung von Molekülkonfigurationen wird in [Pol03] beschrieben. Zum Einsatz kommen hier XML-Stylesheets (XSL-T), welche Daten aus der Chemical Markup Language (CML) nach X3D überführen. Die vielversprechenden Ergebnisse der Arbeit lassen auf eine höhere Akzeptanz für ähnliche modulare Content-Creation-Pipelines hoffen. Im vorliegenden Beitrag werden XML-Stylesheets genutzt um die Aufgabendefinitionen direkt in HTML-Seiten zu transformieren, sowie um Informationen aus X3D Dateien zu extrahieren.

Vielmals werden 3D-Darstellungen für kollaborative virtuelle Lernumgebungen benutzt. Dabei kommen eigenentwickelte, webbasierte Lösungen (z.B. CLEV-R [MMB08]), Kombinationen fremdentwickelter Plattformen wie Second Life und Moodle (z.B. Sloodle [LK08]), oder deklarative Web-Techniken (z.B. DIEL [DCDP10]) zum Einsatz. Der vorliegende Beitrag befasst sich jedoch nicht mit solchen Lernumgebungen, die gewissermaßen einen Virtual-Reality-Ansatz verfolgen. Vielmehr geht es hier um die Darstellung einzelner Objekte, nicht ganzer Welten, sowie die Interaktion mit diesen. Durch Nutzung von Standard-Dateiformaten wird auch der Anpassungsaufwand bestehender Lernmanagement-Systeme minimiert.

3 3D im e-Assessment-Kontext

Der Übergang von Lernmaterialien vom traditionell-zweidimensionalen in den dreidimensionalen Raum bringt viele Möglichkeiten der Wissensvermittlung mit sich, stellt aber auch neuartige Herausforderungen an die Arten und Weisen der Navigation, Interaktion und Präsentation.

3.1 Herausforderungen

In der Regel bieten interaktive dreidimensionale Darstellungen ein objektbezogenes, aber frei rotier- und zoombares Kameramodell. Eine völlig freie Navigation ist besonders im Kontext von e-Assessments problematisch: Wie können die Probanden daran gehindert werden, sich zu verirren, also wie kann zum Beispiel garantiert werden, dass während der Bearbeitungszeit einer Auswahlaufgabe wirklich alle Optionen erkennbar gemacht werden? Die Stärke von 3D, Tiefendaten zu nutzen, um Verdeckungen durch Wahl eines neuen Sichtpunktes umgehen zu können, verkehrt sich hier in ein Hindernis. Die Vorauswahl guter Sichtpunkte durch die Lernobjekt-Autoren sowie die Wahl geeigneter eingeschränkter Navigationsmöglichkeiten ist deshalb sehr wichtig (vgl. [Jan11] zu weiteren Hilfen der Navigation und Wegfindung).

Die Hervorhebung und Kenntlichmachung eines ausgewählten Objekts, der so genannte Fokus, ist für den vorliegenden Anwendungsfall von großer Bedeutung. Bei einheitlich unbunten Modellen kann die Auswahl durch die Einfärbung mit einer Signalfarbe versinnbildlicht werden. Existieren jedoch Objekte mit komplexen Materialien mit Shadern oder Texturen in der Szene, so ist die Wahl einer anderen Methode nötig. Ein Glow-Effekt kann hier effektiv eingesetzt werden, um den Fokus und die Auswahl anzuzeigen, dafür kann z.B. ein Fragment-Shader benutzt werden.

Die Manipulation von 3D-Daten innerhalb der Darstellung ermöglicht Selbstkontrolle nach konstruktivistischen Prinzipien. Als Einschränkung ist hier leider gegeben, dass viele 3D-Darstellungsbibliotheken keine Änderung der Szenenparameter zulassen, vielmehr handelt es sich hier meist nur um Player-Komponenten. Zusätzlich ist bei vielen Anwendungsfällen zu beachten, dass die Nutzergruppe unter Umständen nicht über das computergrafische Fachwissen verfügt, komplexe Manipulationswerkzeuge einzusetzen.

Einfache, aber effektive Manipulationssteuerung ist zum Beispiel möglich über die Anzeige von Widgets an den Szeneteilen, oder eine Drag & Drop-unterstützende Materialbibliothek, wie man sie aus Consumer-Programmen wie Unity3d oder SketchUp kennt. Eine andere Herangehensweise ist die einer eingeschränkten Manipulation, so dass etwa nur an bestimmten Orten in der Szene eine bestimmte Auswahl von Objekten eingefügt werden kann.

3.2 Aufgabentypen

Aus diesen Anforderungen und Möglichkeiten lassen sich Aufgabentypen ableiten, die sinnvoll von der dritten Dimension Gebrauch machen, wobei als Grundlage die weitverbreiteten Aufgabentypen wie Multiple-Choice, Zuordnungsaufgaben, Beschriftungsaufgaben und Freitextaufgaben dienen. Da diese in QTI bereits definiert sind, kann die Erweiterung ins dreidimensionale auf konsistente und logische Weise erfolgen. Für die weitere Untersuchungen wurden die folgenden Aufgabentypen definiert:

Multiple Picking (*q3d:choice3dInteraction*): Die Lernenden können durch Anwählen mittels Mausclick vorher definierte Teile einer 3D-Szene markieren. Es können also Fragen zu Merkmalen eines Objekts gestellt werden, und der Proband kann die entsprechenden Objektteile direkt in ihrem Kontext finden und markieren. In Abbildung 1 ist dieser Aufgabentyp zu sehen.

Freies Labeling (*q3d:labelingInteraction*): Die Lernenden können 3D-Objekte auswählen und in einer Eingabemaske freien Text dazu eingeben. Dieser wird im 3D-Kontext an der entsprechenden Stelle angezeigt. Abbildung 2 zeigt eine Beispielsituation aus dem Prototypen. Komplexere Aufgaben sind denkbar, in denen Studenten der Medizin dreidimensionale anatomische Abbildungen durch Beschriftungen ergänzen sollen.

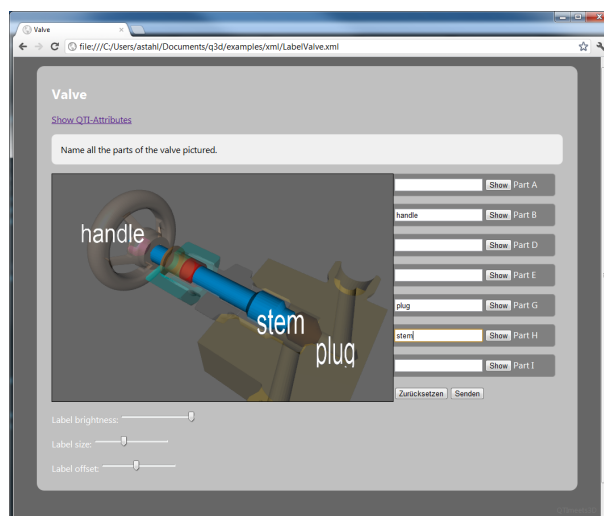


Abbildung 2: Eine Labeling-Aufgabe im Prototypen. Klickt der Nutzer auf ein Objekt, wird das zugehörige Textfeld aktiviert. Die Beschriftung ändert sich in Echtzeit.

Labeling unter Vorgaben (*q3d:labelAssociation3dInteraction*): Die Lernenden können vordefinierten Positionen über eine Auswahlmatrix Wörter und Begriffe zuweisen.

Diese werden in der 3D Ansicht an der entsprechenden Stelle angezeigt. Es handelt sich also um eine geschlossene Variante des freien Labelings.

Missing Item (*q3d:gapMatch3dInteraction*): Die Lernenden können an markierten Lücken in einer 3D-Szene vordefinierte Objektteile über eine Auswahlmatrix einfügen. In Abbildung 3 ist eine solche Aufgabe skizziert. Gezeigt ist ein einfaches Beispiel, es ist aber eine Aufgabe denkbar, in der ein Proband bestimmen soll, welches Bauteil an einer bestimmten Stelle eines Mechanismus fehlt. Dadurch, dass die aktuelle Auswahl im Kontext des restlichen Modells angezeigt wird, kann die Selbstsicherheit des Lernenden unterstützt werden.

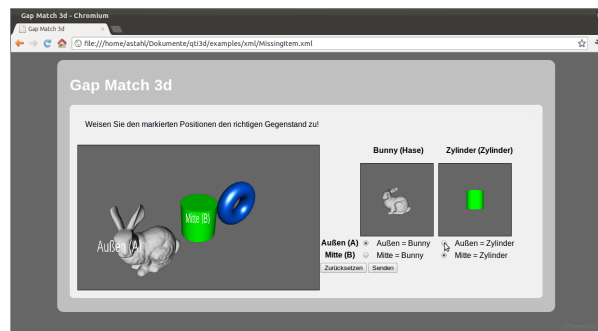


Abbildung 3: Eine Missing-Item-Aufgabe im Prototypen. Die Zuweisung von Objekten zu den Lücken wird über die Matrix auf der rechten Seite vorgenommen. Das Objekt wird in Echtzeit in die 3D-Szene eingefügt. Vorher kann der Nutzer noch die einzufügenden 3D-Objekte in den Vorschauabbildungen interaktiv betrachten.

Point of Interest (*q3d:hotspot3dInteraction*, bzw. *q3d:selectPoint3dInteraction*): Die Lernenden können frei Markierungen in einer Szene setzen. Mittels einfacher Abstandsmessungen kann die Gültigkeit der Antwortkonfiguration ermittelt werden. Dies kann beispielsweise dazu dienen, Wartungspunkte eines Bauteils zu identifizieren oder interessante Orte eines dreidimensionalen Funktionsplots abzufragen.

Im weiteren Verlauf des Beitrags werden Erweiterungen des QTI-Standards präsentiert, die diese neuen Aufgabentypen repräsentieren. Die Erweiterungen beinhalten XSD-Dateien mit XML-Schema-Definitionen für die neuen Interaktionstypen, sowie XML-Stylesheets und JavaScript-Programme zur Erzeugung von prototypenhaften HTML-Seiten direkt aus den Aufgabendefinitionen.

4 Erweiterungen von QTI

QTI erlaubt die Definition von Aufgaben (*AssessmentItem*) und Teststrecken (*AssessmentTest*) mittels eines XML-basierten Datenmodells. Ein *AssessmentItem* enthält dabei meh-

rere Aufgabenteile (*Interaction*) und eine Beschreibung, wie die Antwortvorgaben (*Response*) durch das Lernmanagement-System auf erreichte Punkte abgebildet werden sollen. Interactions stellen den Kernpunkt der hier vorgestellten Erweiterungen dar, da sie die sichtbare Seite der Aufgaben modellieren.

Zum Vorhalten der 3D-Daten bietet es sich an, ein XML-basiertes Datenformat zu benutzen, da die Daten somit nahtlos in das QTI-XML eingefügt werden können. Als XML-Formate für 3D-Daten kommen hauptsächlich X3D und COLLADA in Frage. Dabei ist COLLADA (COLLABorative Design Activity, [Khr08]) als hochflexibles Datenaustauschformat für die 3D-Grafikproduktion konzipiert und bietet einen sehr hohen Funktionsumfang. Eine einfache Möglichkeit der Anzeige direkt in einer Webseite existiert für COLLADA aber nicht.

Da diese gewünscht ist, und viele Funktionen von COLLADA im vorliegenden Kontext nicht benötigt werden, bietet sich das einfachere X3D-Format (Extensible 3D Graphics, [Web08]) an. Mittels der X3DOM-Bibliothek [BEJZ09] können 3D-Objekte nahtlos innerhalb von Web-Seiten dargestellt und die Manipulation der 3D-Szene über die JavaScript-DOM Funktionen des Webbrowsers bewerkstelligt werden.

Wie die einzelnen Teile der Erweiterungen, also Schemadefinition, XSL-T und JavaScript zusammenhängen, ist in Abbildung 4 skizzenhaft dargestellt.

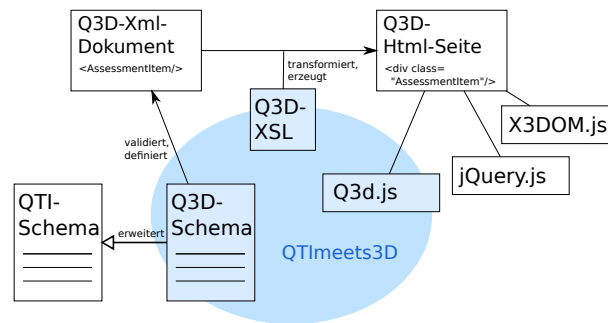


Abbildung 4: Architekturdiagramm der in diesem Beitrag vorgestellten Erweiterungen. Die blau hinterlegten Elemente stellen die Artefakte der QTimeets3D-Erweiterungen dar. Das XML-Schema definiert und validiert QTI Dateien mit 3D-Interaktionen, die wiederum von XSL-T in eine HTML-Seite übertragen werden. Diese greift auf das Q3D-JavaScript zurück, um die Interaktionen zu bewerkstelligen.

4.1 XML-Schema

QTI bietet zur Erweiterung der Interactions die Möglichkeit an, beliebige Daten in einem *CustomInteraction* Objekt einzutragen. Eine *CustomInteraction* ist in dem QTI-Schema so definiert, dass sie beliebigen Inhalt enthalten kann. Dies ist jedoch für Zwecke einer Validierung von XML-Instanzen wenig optimal. Daher wurde in einem zweistufigen Ab-

leitungsprozess zunächst ein *restriction*-Element eingeführt, welches die möglichen Inhalte des CustomInteraction-Typs auf das Notwendigste einschränkt. CustomInteraction wird somit also ähnlich zu einem abstrakten Objekttyp umdefiniert. Von dieser so eingeschränkten *q3dCustomInteraction*-Interaktion werden die anderen hier vorgestellten Interaktionselemente mittels *extension* abgeleitet.

Die QTImeets3D-Erweiterungen (Q3D) übernehmen viele Konzepte direkt aus dem QTI-v2.1-Schema, indem Datentypen, die im QTI-Schema definiert sind, über den Referenz-Mechanismus *xsd:ref="..."* wiederverwendet werden. Dieses Vorgehen ermöglicht eine höhere Kompatibilität mit bestehenden Autoren-Werkzeugen.

Es existieren einige feste Einschränkungen in der QTI-Definition, die eine Verwendung im dreidimensionalen erschweren, da der Standard lediglich von bildbasierten, zweidimensionalen Medieninteraktionen ausgeht. So gibt es zwar vorgefertigte Shape und Area Elemente zur Bereichsdefinition (z.B. als Droptarget), diese beschränken sich aber nur auf Kreise und andere Gebilde in der Ebene. Es wurde von der IMS versäumt, durch dimensionsunabhängige Datendefinition die Erweiterbarkeit von QTI zu vereinfachen. Entsprechend müssen Lernmanagement-Systeme dahingehend programmiert werden, dreidimensionale, volumenförmige Bereichsdefinitionen zu akzeptieren.

Das Einfügen von X3D-Daten in die Q3D-Interaktionen wird in der Schemadefinition über zwei Wege ermöglicht: Direkt als *X3D*-Element oder als *Object*-Element mit dem *type*-Attribut *xml+x3d*. Ersteres entspricht der Behandlung von anderen XML-Datenformaten wie etwa SVG oder MathML innerhalb QTIs, letztere Herangehensweise ist von der allgemeinen Mediendefinition in QTI hergeleitet.

Um einzelne Teile eines 3D-Modells für Interaktionen verfügbar machen zu können, ist es wichtig, dass ein Mechanismus existiert, mit dem sie im Szenengraphen adressiert werden können. Da die Darstellung innerhalb eines Web-Browsers erfolgt, werden CSS-Selektoren für diese Aufgabe benutzt. Diese Wahl ist dadurch begründet, dass CSS ein weitverbreiteter Mechanismus ist, der nativ durch die DOM-Funktion *getElementsBySelector()* in vielen Browsern unterstützt wird. Die Alternative, X-Path, ist trotz ihrer Mächtigkeit im Internet kaum verbreitet.

Es wird ein *q3d:associableSelector*-Element definiert, das beispielhaft folgende Struktur besitzt:

```
<q3d:associableSelector cssSelector=".plug" identifier="A"
  label="Plug"/>
```

In diesem Beispiel wird der CSS-Selector *.plug* benutzt, um eine Menge der Teilbäume des Szenengraphen zu definieren, deren Wurzelemente die XML-Klasse *plug* besitzen. Beim Authoring der X3D Daten sollte also darauf geachtet werden, dass sinnvolle Gruppierungen über Klassen oder eindeutige Identifikatoren mittels CSS adressierbar sind.

Das *identifier*-Attribut des *q3d:associableSelector*-Elements ist für die serverseitige Auswertung der Antwort des Benutzers vorgesehen und entspricht dem gleichnamigen Attribut eines *qti:simpleChoice* Elements. Das *label*-Attribut ist ein optionaler Wert, der zum Beispiel für die Beschriftung der Auswahl-Elemente des Formulars oder innerhalb der Szene benutzt werden kann.

Die Bedeutung des *q3d:associableSelector* ist unterschiedlich, je nachdem in welchem Zusammenhang er verwendet wird. Im Kontext einer *q3d:choice3dInteraction* definiert er ein auswählbares Szenenobjekt, als Element einer *q3d:labeling3dInteraction* ein zu beschriftendes Objekt und bei einer *q3d:gapMatch3dInteraction* wird er als Markierung für den Gap (Lücke), also das zu ersetzende Objekt benutzt. Die Objekte, die stattdessen an der Stelle eingefügt werden können, werden durch *q3d:gapObject3D*-Elemente definiert. Dies entspricht dem Aufbau einer *qti:graphicGapMatchInteraction* im QTI-Standard, die aus *qti:associableHotspots* und *qti:gapImg*-Objekten zusammengestellt wird.

4.2 Referenz-Implementierung

Im Folgenden wird ein auf XML- und HTML5-Technologien aufbauender Prototyp vorgestellt. Dieser dient dazu, um die Interaktionsgestaltung auf der Client-Seite evaluieren zu können. Eine serverseitige Verarbeitung und Speicherung der Antwortdaten übersteigt den Umfang dieser Untersuchung.

4.2.1 XSL-T

Zur Darstellung von *qti:assessmentItems* und *qti:assessmentTests*, den QTI-Elementen zur Definition von Lernkontrollen, innerhalb eines Webbrowsers steht ein XML-Stylesheet (XSL-T) zur Verfügung, welches HTML-Dateien aus den QTI-XML-Quelldaten generiert. Dabei werden für alle Q3D-Interaktionen zunächst HTML-Formular-Elemente erzeugt, die den internen Zustand der Aufgabenstellung enthalten. Weiterhin werden die benötigten JavaScript-Elemente in der HTML-Datei platziert.

Dadurch, dass die Interaktionen auf HTML-Formularen aufbauen, ist es möglich, die Aufgabe auch dann zu lösen, falls die 3D-Anzeige oder JavaScript durch den Browser nicht unterstützt wird.

4.2.2 JavaScript und HTML

Die eigentliche Interaktionslogik wird durch JavaScript realisiert. Das entwickelte Programm nutzt die jQuery-Bibliothek zum Verknüpfen der Formular-Elemente mit den X3D-Elementen (durch Auswertung der CSS-Selektoren in *q3d:associableSelector*) und zur DOM-Manipulation.

Dabei existiert immer eine bidirektionale Verbindung zwischen 3D-Element und HTML-Formular: wird ein 3D-Objekt angeklickt, so wird auch das Klick-Event des zugehörigen Formular-Elements ausgelöst. Ändert sich der Zustand des Formular-Elements, ändert sich das 3D-Objekt entsprechend, zum Beispiel ändert ein selektiertes Objekt die Farbe. Wird der Inhalt eines Textfelds geändert, so spiegelt sich das umgehend in einer Änderung der Beschriftung im 3D-Kontext wider.

Durch diese Verknüpfung zwischen Datenmodell und Ansichten kann die dreidimensionale Darstellung als eine Illustration des Bearbeitungszustands angesehen werden, welcher

auch über herkömmliche HTML-Elemente manipulierbar ist. Diese alternative, indirekte Manipulation gewährleistet, dass die Probanden immer alle Lösungsmöglichkeiten wählen können, auch wenn sie an der Navigation in der Szene scheitern, falls etwa keine geeigneten Sichtpunkte definiert worden sind.

Eine ähnliche Problematik tritt auch im Zusammenhang mit der Barrierefreiheit auf. Damit eine HTML-Seite von Screenreader-Geräten und Braille-Zeilen ausgegeben werden kann, dürfen nur Standardkonforme HTML-Elemente benutzt werden. Weiterhin müssen grafische Elemente sinnvoll annotiert werden. Die Form-Input-Elemente als Träger des Datenmodells zu benutzen, ermöglicht es Bildschirmleseprogrammen, die Interaktions- und Antwortmöglichkeiten zu erkennen.

Dadurch, dass X3DOM die WebGL-Schnittstelle benutzt, beschränkt sich die Nutzergruppe auf jene mit bestimmten Browsern. Sowohl Mobilgeräte als auch der Internet Explorer von Microsoft lassen bisher aufgrund von Sicherheitsbedenken den GPU-Zugriff von Webseiten mittels WebGL nicht zu. X3DOM bietet aber die alternative Möglichkeit der Anzeige mittels eines Flash-Plugins, womit auch diese Browser unterstützt werden können.

5 Diskussion der Ergebnisse

Die präsentierte Anwendungsarchitektur ist sehr flexibel und kann durch die Verwendung von Webstandards mit wenig Aufwand in bestehende Lernmanagement-Systeme adaptiert werden. Der Server muss lediglich die Verarbeitung von XSL-Transformationen unterstützen, um die HTML-Fragmente zu generieren, ohne dem Client Zugriff auf die QTI-Quellen zu geben. Die Ergebnisdaten entsprechen weitestgehend den in QTI definierten, weshalb kaum Anpassungen an der Auswertungsinfrastruktur notwendig sind.

Bisher ist für die 3D-Daten und die Erstellung der Q3D-Aufgaben kein Authoring-Prozess definiert. Die Fähigkeit zum Export von X3D ist nur von wenigen Tools umgesetzt. Als Zwischenschritt bietet sich das ältere VRML-Format an, welches verlustlos in X3D mittels den Werkzeugen des InstantReality-Toolkit (www.instantreality.org) umgewandelt werden kann.

Es sind weitere Aufgabentypen denkbar, wie etwa eine *PositionObject3dInteraction*. Hierzu muss es jedoch möglich sein, komplexe Positionierung von Objekten interaktiv und einfach zu gestalten, etwa unter Verwendung von Manipulations-Widgets, wie sie aus CAD und anderen 3D-Grafikprogrammen bekannt sind. Bisher bietet X3DOM diese Manipulationsmöglichkeit noch nicht selbst, sie kann aber über JavaScript nachgerüstet werden.

Das nächste Ziel des Projekts ist die Integration in gängige Lernmanagement-Systeme und QTI-Player-Software, wie etwa die ONYX-Suite der BPS Bildungsportal Sachsen GmbH. Damit im Zusammenhang steht eine ausgiebige Evaluation der neu gewonnenen Interaktionsmöglichkeiten, sowohl im Authoring- als auch im Nutzer-Einsatz. Hierfür wird unter anderem die Akzeptanz und Verbreitung von Web-3D-Technologien in der nahen Zukunft eine Rolle spielen.

Weiterer Betrachtung bedarf auch die Präsentation von Feedback zu gelösten Aufgaben, zum Beispiel wie der Vergleich zu einer Musterlösung im Dreidimensionalen zu gestalten ist. Da dieses Mittel zur Selbstkontrolle im Kontext von e-Assessments extrem wichtig ist, sollten entsprechende Vorkehrungen, die bereits in QTI existieren, für die hier vorgestellten Interaktionen umgesetzt werden.

5.1 Zusammenfassung

Durch Erweiterungen der XML-Schema-Definitionen des e-Assessment-Standards QTI v2.1 ist es möglich, verschiedene Aufgabentypen zur Abfrage von Wissen anhand von dreidimensionalen Daten zu verwenden. Die bisher entwickelten Aufgabentypen leiten sich von den üblichen Interaktionsmöglichkeiten eines 3D-Web-Interfaces sowie herkömmlichen Aufgabentypen wie Multiple-Choice- und Zuordnungsaufgaben ab. Es sind aber verschiedene weitere Aufgabentypen denkbar.

Eine Referenzanwendung, die vollkommen auf Web-Standards wie XML, XSL-T, HTML und JavaScript aufbaut, ermöglicht die Erprobung der Spezifikation. Dabei wird auf eine hohe Kompatibilität mit verschiedenen Anzeigegeräten geachtet, indem jeder Aufgabe eine HTML-Form zugrunde liegt, die zu jedem Zeitpunkt den Zustand der Fragebearbeitung enthält.

6 Danksagungen

Diese Publikation wurde gefördert durch die Mittel des Multimediafonds des Medienzentrums der TU-Dresden im Förderschwerpunkt *Maßnahmen zur Umsetzung mediengestützter Prüfungen*.

Literatur

- [BEJZ09] Johannes Behr, Peter Eschler, Yvonne Jung und Michael Zöllner. X3DOM: a DOM-based HTML5/X3D integration model. In *Proceedings of the 14th International Conference on 3D Web Technology, Web3D '09*, Seiten 127–135, New York, NY, USA, 2009. ACM.
- [CR07] Luca Chittaro und Roberto Ranon. Web3D technologies in learning, education and training: Motivations, issues, opportunities. *Comput. Educ.*, 49(1):3–18, August 2007.
- [DCDP10] Francesco Di Cerbo, Gabriella Dodero und Laura Papaleo. Integrating a Web3D interface into an e-learning platform. In *Proceedings of the 15th International Conference on Web 3D Technology, Web3D '10*, Seiten 83–92, New York, NY, USA, 2010. ACM.
- [Huk06] T. Huk. Who benefits from learning with 3D models? the case of spatial ability. *Journal of Computer Assisted Learning*, 22(6):392–404, 2006.

- [Huk07] Corrigendum for original article by Huk, T. (2006). *Journal of Computer Assisted Learning*, 23(2):169–169, 2007.
- [Hö10] Tim Höfler. Spatial Ability: Its Influence on Learning with Visualizations—a Meta-Analytic Review. *Educational Psychology Review*, 22:245–269, 2010.
- [IMS06] IMS Global Learning Consortium, inc. *IMS Question & Test Interoperability Specification Public Draft v2.1 (revision 2)*, Juni 2006. <http://www.imsglobal.org/question/> (last visited on 29th of June 2012).
- [Jan11] Jacek Jankowski. A taskonomy of 3D web use. In *Proceedings of the 16th International Conference on 3D Web Technology*, Web3D '11, Seiten 93–100, New York, NY, USA, 2011. ACM.
- [Khr08] Khronos Group, Sony Computer Entertainment Inc., Clearlake Park, CA 95424, U.S.A. *COLLADA – Digital Asset Schema Release 1.5*, April 2008. <http://www.khronos.org/collada/> (last visited on 29th of June 2012).
- [LK08] D Livingstone und J Kemp. Integrating Web-Based and 3D Learning Environments: Second Life Meets Moodle. *UPGRADE The European Journal for the Informatics Professional*, IX(3):8–14, 2008.
- [MMB08] T. Monahan, G. McArdle und M. Bertolotto. Virtual reality for collaborative e-learning. *Computers & Education*, 50(4):1339–1353, Mai 2008.
- [Pol03] Nicholas F. Polys. Stylesheet transformations for interactive visualization: towards a Web3D chemistry curricula. In *Proceedings of the eighth international conference on 3D Web technology*, Web3D '03, Seiten 85–ff, New York, NY, USA, 2003. ACM.
- [SPC⁺09] Florin D. Salajan, Susanne Perschbacher, Mindy Cash, Reena Talwar, Wafa El-Badrawy und Greg J. Mount. Learning with web-based interactive objects: An investigation into student perceptions of effectiveness. *Computers & Education*, 53(3):632 – 643, 2009.
- [Web08] Web3DConsortium. *X3D Architecture and base components Edition 2*, 2008. <http://www.web3d.org/x3d/specifications/> (last visited on 29th of June 2012).
- [YN04] Andy J. Yeh und Rodney A. Nason. Knowledge Building of 3D Geometry Concepts and Processes within a Virtual Reality Learning Environment. In *World Conference on Educational Multimedia, Hypermedia & Telecommunications*, Lugano, Switzerland, 2004. Association for the Advancement of Computing in Education (AACE).
- [Çe12] Aydın Çetin. 3D Web Based Learning of Medical Equipments Employed in Intensive Care Units. *Journal of Medical Systems*, 36:167–174, 2012.