

# Interaktives Notfalltraining in der Zahnmedizin: Entwicklung einer Virtual Patient Player- Simulationssoftware

## Interactive emergency training in dentistry: Development of a virtual patient player simulation software

### Abstract

Virtual patients have long been established as part of medical training. With their help, students can develop their therapeutic and diagnostic skills under conditions that are as realistic as possible, without any risk for a patient. While model or dummy simulations represent a standard in dentistry for the training of manual skills, computer-based, interactive simulations of virtual patient cases in this field are rarely described. Some of these systems have been designed for a limited purpose and do not permit a flexible and reusable design of virtual patients, or only permit it with restrictions. The MedBiquitous standard offers solutions for this purpose. This has proven itself in the medical field as a basis for virtual patient simulations. It offers an expandable structure for the data of virtual patients and describes fundamental algorithms of associated simulation software. In this paper, virtual patient player simulation software for interactive emergency training in dentistry is developed on the basis of the MedBiquitous model. As an example, a first virtual patient was created. MedEmergency Player enables action-focussed simulation without rigidly predefined linear or branched treatment pathways, unlike existing systems and the previous use of the MedBiquitous standard. The user directly experiences and examines the reactions as with a real patient

**Keywords:** emergency, dentistry, MedBiquitous, virtual patients, simulation

### Zusammenfassung

Virtuelle Patienten sind in der medizinischen Ausbildung seit langem etabliert. Mit ihrer Hilfe können Lernende ohne Risiko für einen Patienten ihre therapeutischen und diagnostischen Fähigkeiten unter möglichst realitätsnahen Bedingungen ausbauen. Während Modell- oder Puppensimulationen in der Zahnmedizin zum Training manueller Fähigkeiten einen Standard darstellen, werden computerbasierte, interaktive Simulationen virtueller Patientenfälle für dieses Fachgebiet selten beschrieben. Einige dieser Systeme sind für einen begrenzten Einsatzzweck konzipiert und lassen eine flexible und wiederverwendbare Gestaltung virtueller Patienten nicht oder nur eingeschränkt zu. Lösungsansätze hierzu zeigt der MedBiquitous-Standard. Dieser hat sich im medizinischen Bereich als Grundlage virtueller Patientensimulationen bewährt. Er bietet eine erweiterbare Struktur für die Daten virtueller Patienten und beschreibt grundlegende Algorithmen einer dazugehörigen Simulationssoftware. In der vorliegenden Arbeit wird auf Grundlage des MedBiquitous-Modells eine Virtual Patient Player-Simulationssoftware für ein interaktives Notfalltraining in der Zahnmedizin entwickelt. Ein erster virtueller Beispielpatient wurde erstellt. Der MedEmergency-Player ermöglicht im Unterschied zu vorhandenen Systemen eine aktionsbeton-

### Martin Geneit<sup>1</sup>

1 Praxis für Zahnmedizin und Oralchirurgie Dr. med. dent. Martin Geneit, Bad Essen, Deutschland

te Simulation ohne „starr“ vorgegebene Behandlungspfade. Der Benutzer erfährt und untersucht die Reaktionen unmittelbar wie bei einem realen Patienten.

**Schlüsselwörter:** Notfall, Zahnmedizin, MedBiquitous, virtuelle Patienten, Simulation

## Einleitung

Bei Notfallsituationen während der zahnärztlichen Behandlung muss das Behandlungsteam angemessen und koordiniert reagieren. Die hierfür nötige notfallmedizinische Ausbildung erwirbt der Zahnarzt während des Studiums und in vertieft sie in Weiterbildungen. Für ein ausführliches, regelmäßiges Üben fehlt allerdings oft ein der Sache angemessenes Zeitkontingent.

Durch regelmäßiges Anwenden prägen sich Lerninhalte des zahnärztlichen Alltags ein. Dies ist bei Notfallsituation nicht möglich. Es sind selten eintretende, plötzliche Ereignisse mit hohem Risiko. Sie können nicht durch einen realen Patientenfall trainiert werden.

Das fachliche Können muss daher auf andere Weise aufrechterhalten werden. Wichtig ist hierbei das rasche Reagieren im „theoretischen“ Bereich, aber auch das regelmäßige Üben der praktischen Handgriffe.

Zu diesem Zweck werden seit Langem virtuelle Patienten im Sinne von nicht computerbasierten naturgetreuen Puppensimulationen [1] verwendet, z.B. bei der Herzlungen-Wiederbelebung. Eine andere Form virtueller Patienten könnte als „interaktive, realitätsnahe, computerbasierte Simulation“ [2] die Notfallsausbildung unterstützen.

In der Zahnmedizin sind solche computergestützten interaktiven Patientenszenarios grundsätzlich bekannt. Das ODSDM/DDx&Tx-System wurde bereits in den achtziger Jahren des 20. Jahrhunderts über einen langen Zeitraum in der studentischen Ausbildung verwendet [3]. Simuliert wurden hier zunächst Patientenfälle aus dem Bereich der oralen Pathologie, später auch Szenarien aus dem gesamten zahnmedizinischen Fachgebiet [4].

Trotz dieser frühen Beschreibung wurden in den folgenden Jahrzehnten in der Zahnmedizin nur wenige computergestützte virtuelle Patienten in der Literatur erwähnt [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]. Im Bereich des Notfalltrainings ist nach Literaturrecherche und dem Kenntnisstand des Autors bisher keine Software bekannt.

Mögliche Ursachen für die geringe Verbreitung sind u.a. hoher Aufwand und Kosten bei der Entwicklung und Anpassung der virtuellen Patienten und die fehlende Wiederverwendbarkeit bzw. Austauschmöglichkeit der Inhalte [15].

Einen solchen wichtigen systemübergreifenden Austausch virtueller Patienten ermöglicht der Standard nach ANSI/MedBiquitous. Er beschreibt eine gemeinsame XML-basierte Struktur für Inhalt und Verhalten der virtuellen Patienten. Dadurch kann eine der Spezifikation entsprechende Virtual Patient Player-Software diese virtuellen

Patienten darstellen. Zudem können Inhalte leichter angepasst und erweitert werden [16].

Der MedBiquitous-Standard wurde bisher hauptsächlich für Virtual Patient Systeme mit dem Schwerpunkt auf textbasierter Entscheidungsfindung eingesetzt. Der Handlungsablauf der Simulationen ist zumeist verzweigt oder linear vorgegeben. Seltener wird auch eine sogenannte problemlösungsorientierte/globale Struktur implementiert, bei der z.B. Anamnese- und Befundungsoptionen ohne vorgegebene Reihenfolge frei ausgewählt werden können [17].

Die Auswahl der Handlungsoptionen erfolgt in jedem Fall durch die aktive Navigation des Benutzers, z.B. per Mausklick. Dieser Ablauf könnte durch mehrere gleichzeitig frei wählbare Handlungsoptionen mit direkter Reaktion des virtuellen Patienten erweitert werden, so dass eine eher „aktionsbetonte“ Software resultiert.

Dieser Artikel beschreibt die Verwendung und Adaption des MedBiquitous-Standards für die unmittelbare Beeinflussung der Parameter eines virtuellen Patienten. Mit der angepassten Struktur wird eine Virtual Patient Player-Software für das interaktive Notfalltraining in der Zahnmedizin entwickelt. Die Software erlaubt auch den Einsatz linearer oder verzweigter Pfadstrukturen.

Leichter weiterverwenden und anpassen lässt sich ein solches System nicht nur durch den Einsatz des MedBiquitous-Standards, sondern auch durch eine plattformunabhängige browserbasierte Frontend-Komponente und die Trennung von Programmlogik, Daten und Benutzeroberfläche.

In die grafische Benutzeroberfläche lassen sich optional multimediale Inhalte einbinden, um die Immersion des Benutzers in das Simulationsgeschehen zu fördern. Zur Demonstration wird ein Prototyp (einfacher Patientenfall) zur simulierten Behandlung eines bei der zahnmedizinischen Behandlung auftretenden Notfalls erstellt.

## Material und Methoden

Die Software wurde mit den in den folgenden Abschnitten angegebenen Hilfsmitteln ohne externe Dienstleister eigenständig durch den Autor entwickelt.

### XML-Daten des virtuellen Patienten

Die XML-Daten der virtuellen Patienten des MedEmergency-Players werden auf einem Server vorgehalten und sind nicht allzu umfangreich. Daher spielen Performance-Unterschiede verschiedener Datenbankmanagementsysteme (DBMS) eine untergeordnete Rolle. Somit kann statt eines nativen XML-DBMS ein XML-

enabled DBMS (Microsoft SQL Server/Visual Studio) eingesetzt werden. Dies ist in Hinblick auf Datenintegrität, Abfragen, nebenläufigen Transaktionen, Standardisierung und Administration vorteilhaft.[18].

Die abschnittsweise Weitervermittlung der vom Server aufbereiteten XML-Daten an das Flex-Frontend erfolgt über dynamische Webseiten des zum SQL-Server „passenden“ Mediators ASP.NET in Verbindung mit MS Visual Basic.

## MedEmergency-Player-Frontend

Das Frontend des MedEmergency-Players soll ein im Vergleich zu herkömmlichen Webanwendungen ein verbessertes Nutzererlebnis ähnlich einer dynamischen Desktopanwendung bieten. Dem Benutzer stehen umfangreiche Darstellungs- und Interaktionsmöglichkeiten wie Animationen, Video, komplexe Kontrollelemente zur Verfügung. Ein kleinerer Teil der Anwendungslogik wird in das Frontend ausgelagert, so dass nicht zu umfangreiche Inhalte vom Server geladen werden müssen. Damit sind einige charakteristische Merkmale einer sogenannten „Rich Internet Application“ (RIA) erfüllt, wobei dieser Begriff weder standardisiert noch exakt definiert ist [19]. Zur Entwicklung des RIA-Frontends wurde das Adobe/Apache Flex Framework mit der IDE Adobe Flash Builder ausgewählt. Dieser bietet bei eingeschränkten Ressourcen im Vergleich zu den Möglichkeiten von HTML/JavaScript eine „Komplettlösung“ [20] zur Erstellung eines datenzentrierten und umfangreicheren Frontends.

Das Flex SDK der Flash-Plattform beinhaltet eine große Anzahl anpassbarer UI Komponenten. Dadurch wird der Programmieraufwand für die Benutzeroberfläche des MedEmergency-Players deutlich verringert. Die Design View des Flash Builders hilft durch ihre visuelle Darstellung bei der Gestaltung und Adaption der Benutzeroberfläche. Design und Layout der Benutzeroberfläche können „pixelgenau“ ohne Berücksichtigung verschiedener Browser(-versionen) erstellt werden. Das Testen der Software erfordert dadurch weniger Aufwand im Vergleich zu HTML/JavaScript.

Erstellte Anwendungen können mit dem Adobe Flash Player in Browsern und mit der plattformübergreifenden Laufzeitumgebung Adobe AIR nach Installation browserunabhängig auch ohne Internetverbindung auf lokalen Desktop- und mobilen Rechnern ausgeführt werden [20]. Der MedEmergency-Player kann somit je nach Einsatzzweck webbasiert gleichzeitig vielen Nutzern zur Verfügung stehen oder auch komplett als lokales System auf einzelnen Rechnern installiert werden.

ActionScript als Programmiersprache der Flash-Umgebung bietet u.a. klassenbasierte Objektorientierung und XML als nativen Datentyp für die Verarbeitung der Inhalte des MedBiquitous-Modells. Für die Sprachausgabe des virtuellen Patienten wird ein Text-to-Speech-System genutzt (Voice RSS: <http://www.voicerss.org/>).

## PureMVC-Framework

Das Frontend des MedEmergency-Players soll möglichst einfach an unterschiedliche Fachgebiete und Lernkonzepte angepasst werden können. Daten, Benutzerschnittstelle und Programmlogik werden daher möglichst unabhängig voneinander gehalten. So kann z.B. die Benutzeroberfläche an ein anderes Fachgebiet ohne aufwändige Anpassungen der anderen Programmteile adaptiert werden. Für diese Anforderungen lässt sich das PureMVC-Framework als Mikroarchitektur einsetzen, da es auf dem MVC-Entwurfsmuster basiert. Änderungen der Software und des komplexen MedBiquitous-Modells werden durch die Aufteilung in **Model** (Daten), **View** (Benutzerschnittstelle) und **Controller** (Programmlogik) vereinfacht. PureMVC ist für umfangreichere Anwendungen geeignet, die wie die MedEmergency Frontend-Komponente auf ActionScript basieren. Die Kommunikation zwischen den Hauptakteuren läuft nach festgelegten Regeln ab, hierbei werden u.a. Notifications als ein von den Flash Event-Klassen unabhängiges ergänzendes Beobachter-Benachrichtigungsmodell (Observer Pattern) verwendet [21].

Der Entwurf innerhalb des PureMVC-Frameworks erfolgt nach der von Hall vorgeschlagenen Vorgehensweise [21]: Zunächst werden Value Objects mit den Entitäten des XML-Modells und View Components für das GUI entworfen und getestet. Diese Bestandteile des Modells sind unabhängig vom Rest des Systems und dienen als Gerüst für den restlichen Aufbau der Applikation.

Anschließend werden die Proxies für den Zugriff auf die Value Objects definiert, dann die Mediators für die „Vermittlung“ zwischen View Components und dem restlichen System. Die Implementierung der Commands zur Steuerung der Programmlogik erfolgt als letzter Schritt.

## MedBiquitous-Standard

Die Grundlage für das Datenmodell des MedEmergency-Players ist der Standard nach MedBiquitous, der neben den Datenstrukturen [22] auch Vorgaben für eine korrespondierende Virtual Patient Player-Software beschreibt [23]. Das MedBiquitous-Modell besteht im Wesentlichen aus vier Hauptbestandteilen [22]: ActivityModel, DataAvailabilityModel, VirtualPatientData und Manifest. Diese werden als getrennte XML-Dokumente codiert. Virtual Patient Data (VPD) beinhaltet ähnlich einer elektronischen Krankenakte die patientenbezogenen Daten des virtuellen Patienten wie z.B. Medikation, Therapiemaßnahmen etc. Media Resources (MR) verweisen auf multimediale Inhalte, während das Data Availability Model (DAM) seinerseits auf VPD- und MR-Elemente verweist und Inhalte für das Activity Model (AM) in einzelnen Knoten (DAM-Nodes) bündelt. Das AM stellt u.a. Handlungsoptionen für den Benutzer mittels einer Knotenstruktur zur Verfügung. Die Knoten (Activity Nodes) sind miteinander verlinkt und enthalten Verweise auf DAM-Nodes. DAM-Nodes, VPD- und MR-Elemente können durch verzweigte Verbindungen (XPath-Verweise) mehrfach benutzt

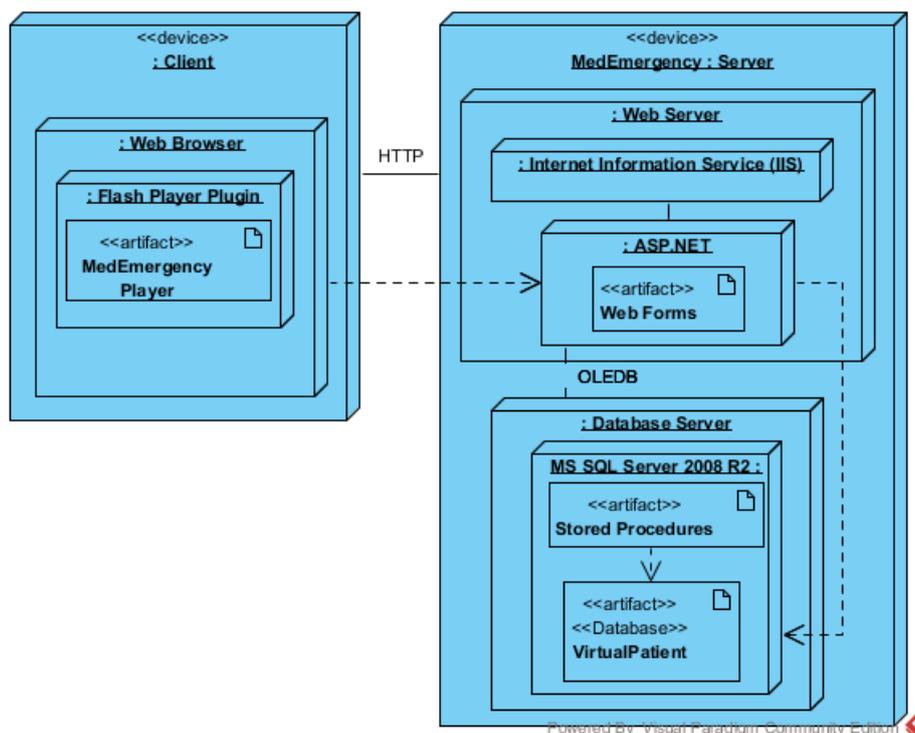


Abbildung 1: Verteilungsdiagramm MedEmergency-Player-System

werden, so dass die systembedingte Redundanz der XML-Daten reduziert wird.

Der Algorithmus eines Virtual Patient Player nach MedBiquitous-Spezifikation besteht aus festgelegten Einzelschritten [23]. Für den „Spielcharakter“ der Simulation sind hierbei u.a. die Strukturen Counter, CounterRules und CounterActionRules von Bedeutung.

Ein Counter repräsentiert einen einzelnen numerischen Wert, der sich abhängig von der Aktion des Benutzers verändern kann. So können beispielsweise Werte für die „Vital-/Simulationsparameter“ des Patienten (Blutdruck, Puls, „Gesundheit“ o.ä.) hinterlegt werden.

In einem Bereich des Activity Models werden die grundlegenden Eigenschaften der Counter festgelegt (ID, Bezeichnung, initialer Wert, Suffix/Präfix) und die Regeln für die vom Counter abhängigen Reaktionen des Programms definiert (CounterRules).

Die einem Counter zugeordneten CounterRules können eine oder mehrere Regeln (Rule) enthalten. In einer Rule ist relationaler Operator („Relation“) und ein Zahlenwert („Value“) codiert. Ist der aus diesen Elementen und dem aktuellen Wert des Counters zusammengesetzte Ausdruck wahr, wird der im Element „RuleRedirect“ als Pfad hinterlegte ActivityNode angesteuert. Im MedBiquitous-Modell werden die Regeln ausschließlich bei einem Wechsel des ActivityNode überprüft.

Im MedBiquitous-Modell wird der Wert eines Counters bei Ansteuerung eines bestimmten Activity-Nodes verändert. Dies wird durch das Element „CounterActionRule“ des einzelnen ActivityNode definiert. Der CounterOperator bestimmt dabei die Art der Wertänderung (z.B. Addition), der CounterRuleValue enthält einen Operanden bzw. einen zuzuweisenden Wert. Im CounterPath befindet sich

ein Verweis im XPath-Format auf den zu ändernden Counter.

Die „Weiterschaltung“ zu einem neuen ActivityNode erfolgt nach MedBiquitous-Standard erst durch dezidiertes Anklicken eines Navigationselements bzw. nach Überprüfung entsprechender Bedingungen vor Eintritt in einen Activity Node.

## Ergebnisse

Mit dem MedEmergency-Player lassen sich Aspekte der Notfallbehandlung eines Patienten während einer zahnmedizinischen Behandlung simulieren. Hierzu kann der Benutzer die Anamnese, Diagnose und Therapie durchführen, der virtuelle Patient gibt daraufhin entsprechende Rückmeldungen in textlicher oder audiovisueller Form. Die Architektur des MedEmergency-Player-Systems wird zusammenfassend anhand eines Verteilungsdiagramms in Abbildung 1 dargestellt.

## XML-Daten des virtuellen Patienten

Die XML-Daten eines virtuellen Patienten werden gemäß dem Verteilungsdiagramm in Abbildung 1 in Tabellenform in einem XML-enabled DBMS gespeichert. Aus den einzelnen getrennten Bestandteilen der Datenstruktur werden relevante Abschnitte zunächst durch den SQL Server über Stored Procedures extrahiert, teilweise aufbereitet und neu zusammengestellt. Somit können für die Weiterverarbeitung „fertig vorbereitete“ XML-Inhalte bereitgestellt werden.

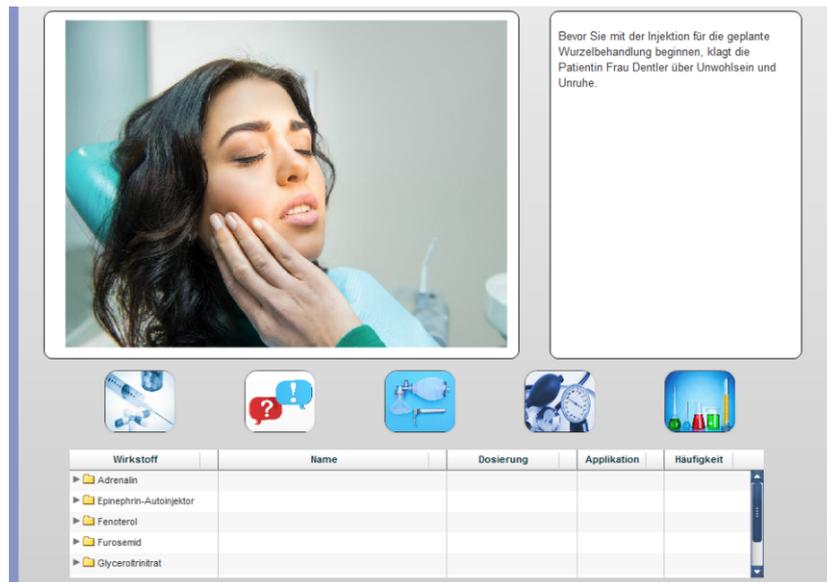


Abbildung 2: GUI MedEmergency-Player mit Icons für Diagnose und Therapie, Untermenü Medikation

Die vom Server bereitgestellten XML-Daten werden anschließend an die Frontend-Komponente mittels HTTPService weitervermittelt. Das Ergebnisformat e4x ermöglicht hierbei eine direkte Weiterverarbeitung durch ActionScript-Methoden. Auf die zur Laufzeit unveränderlichen XML-Inhalte wird mit dem REST-Architekturstil zugegriffen.

Die im Web-Browser mit Hilfe des Flash Player-Plugins betriebene Frontend-Komponente verarbeitet die erhaltenen Daten durch Algorithmen der (angepassten) MedBiquitous Player-Spezifikation. Die Interaktion mit dem virtuellen Patienten erfolgt über eine grafische Benutzeroberfläche mit integrierter Sprachausgabe.

## MedEmergency-Player-Frontend

Bei der Gestaltung der Benutzeroberfläche wurde ein möglichst übersichtlicher Aufbau ohne Überfrachtung mit zu vielen Programmsymbolen gewählt. Somit ist auch ein Einsatz auf mobilen Geräte (Smartphone, Tablet) einfach möglich. Die Flex-Standardkomponenten (Buttons, TextAreas etc.) wurden in Ihrer Gestaltung modifiziert (abgerundete Kanten, Buttons mit semitransparenten Icons etc.). Der Bildschirmaufbau wurde in drei Bereiche aufgeteilt. Abbildung 2 zeigt dies anhand eines implementierten Beispielpatienten:

- ein Feld für Multimediakomponenten (Grafiken, Animationen etc.) u.a. für die Möglichkeit bildhafter Darstellung der Behandlungssituation,
- ein Textfeld für die Ausgabe von Patientenreaktionen, Befunden, Bewertungen und sonstigen Programmmeldungen,
- die Menüleiste mit Icons für Medikation, Interviewfragen, nicht-medikamentöse Therapie (z.B. Beatmung), Labor- und physikalischen Untersuchungen (z.B. Blutzucker-/Blutdruckmessung).

Ein Untermenü mit Handlungsoptionen erscheint als (Advanced)DataGrid durch Anklicken der Buttons. Bei Auswahl von Interviewfragen erfolgt eine Ausgabe im Textfeld und zusätzlich eine Sprachausgabe der Antwort des Patienten. Die Menüpunkte werden in der XML-Struktur des virtuellen Patienten festgelegt und lassen sich somit flexibel an die Aufgabenstellung anpassen (XML-Elemente des Bereichs VirtualPatientData des Modells, z.B. InterviewItem, Medication, DiagnosticTest etc.).

## PureMVC-Framework

Das Frontend verwendet das PureMVC-Framework [21] und ist entsprechend dem Meta-Entwurfsmuster MVC in drei Pakete „model“, „view“ und „controller“ unterteilt: Das Paket „model“ enthält Proxy-Klassen für die Bereitstellung und Speicherung von (XML)-Daten des virtuellen Patienten (z.B. ActivityNodes, Links, Counter etc.). Für die temporäre Datenspeicherung werden „value object“-Klassen benutzt.

Das Paket „view“ definiert die grafische Benutzeroberfläche des MedEmergency-Players durch die Klassen der View Components. Den View Components sind Mediator-Klassen zugeordnet. Eine View Component kann über eine Event-Klasse Nachrichten an eine ihr zugehörige Instanz einer Mediator-Klasse senden. Das Mediator-Objekt verbindet wiederum die zugeordnete View Component durch bidirektionalen Datenaustausch mit dem Rest der Applikation.

Im Paket „controller“ befinden sich Command-Klassen zur Steuerung der Programmlogik und zur Initialisierung des PureMVC-Frameworks. Command-Klassen können Zugang zu den Klassen der Pakete „view“ und „model“ haben.

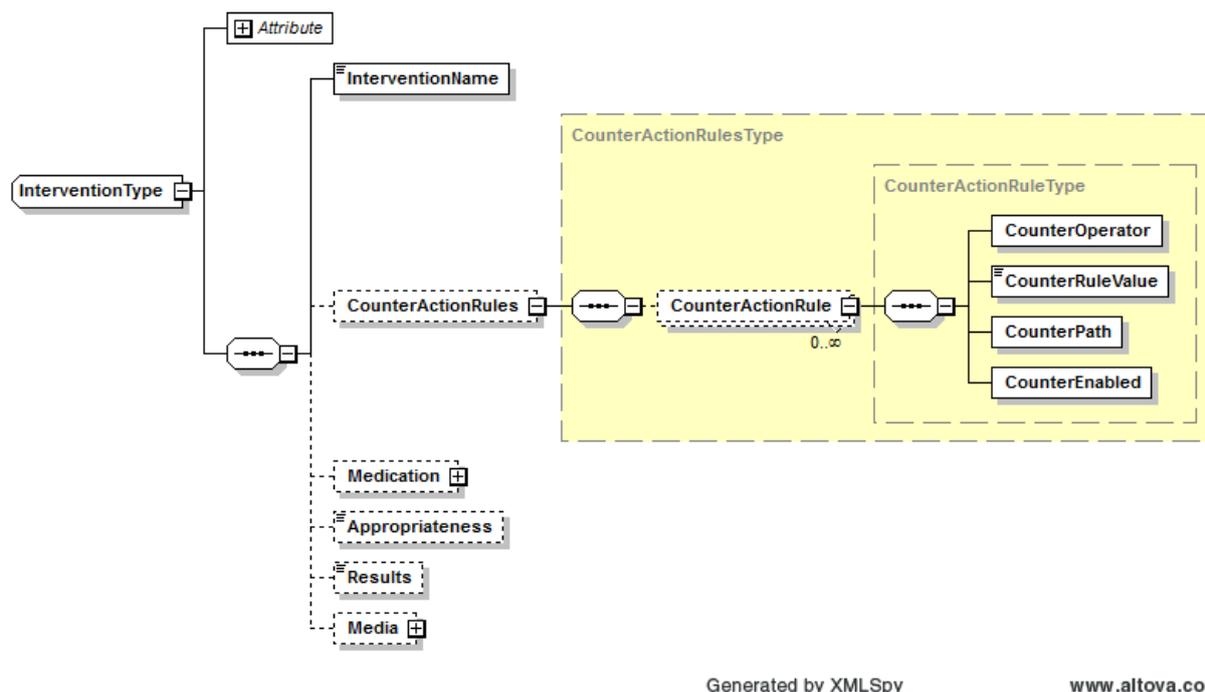


Abbildung 3: Erweiterung Schema durch CounterActionRule

## MedBiquitous-Standard

Der handlungsorientierte MedEmergency-Player ermöglicht eine optionale direkte Weiterschaltung nach einer oder mehreren vorher festgelegten Benutzerinteraktionen mit dem VPD-Bereich des Modells (z.B. Medikamentengabe) zum nächsten ActivityNode.

Parameter des virtuellen Patienten (z.B. der Blutdruck) können ohne zwingende Rückmeldung verändert werden. So würde z.B. der Blutdruck durch ein Medikament noch nicht ausreichend gesenkt; erst durch eine weitere Dosis wird ein Wert der Counter-Variablen erreicht, der zu einem anderen ActivityNode mit entsprechender Rückmeldung führt.

Daher werden Counter nicht nur bei der Behandlung von ActivityNodes anhand von CounterRules geprüft und durch CounterActionRules verändert, sondern auch bei einer Interaktion des Benutzers mit VPD-Elementen. Die Syntax der im Bereich der VPD-Elemente ergänzten CounterActionRules ist analog zu den ActivityNode-Abschnitten aufgebaut (s. Abbildung 3).

## Diskussion

Im Bereich des zahnmedizinischen Notfalltrainings ist der unterstützende Einsatz virtueller Patienten als computergestütztes interaktives Patientenszenario naheliegend: Studenten und Zahnärzte könnten mit ihrer Hilfe gefahrlos Diagnose- und Therapieaspekte ohne Angst vor Fehlern erlernen, da reale Notfallpatienten zu Übungszwecken nicht verfügbar sind. Im Bereich der Notfalltherapie ist nach Literaturrecherche und dem Kenntnisstand des Autors jedoch bislang kein derartiges System bekannt.

Der in dieser Arbeit vorgestellte MedEmergency-Player simuliert mit Hilfe eines solchen virtuellen Patienten Aspekte der Notfallbehandlung während einer zahnmedizinischen Therapie. Hierbei kann der Benutzer mit Hilfe der Software notwendige schnelle Entscheidungen als Ergänzung zu praktischen Übungen trainieren. Die Programmfunktionen bei Diagnose und Therapie eines solchen simulierten Behandlungszwischenfalls verdeutlicht ein für das System beispielhaft erstellter Notfallpatient. Virtuelle Patienten als computergestützte interaktive Patientenszenarios werden in der Zahnmedizin selten beschrieben [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]. Bei der Entwicklung des MedEmergency-Players sollten daher in der Literatur beobachtete „Stolpersteine“ realisierter Simulationen möglichst vermieden werden, die den Einsatz virtueller Patienten behindern: Virtuelle Patienten zu erstellen ist aufwändig und zeitintensiv, vorhandene Ressourcen müssen effektiv eingesetzt werden. Ein Standard wie der nach ANSI/MedBiquitous kann diese Aufgabe erleichtern: Durch vorgegebene Datenstrukturen und Algorithmen eines Virtual Patient Player lassen sich virtuelle Patienten einfacher erstellen, anpassen und wiederverwenden [15]. Nach Literaturrecherche und Kenntnisstand des Autors wird lediglich für das Web-SP System die Implementierung des MedBiquitous-Standards beschrieben [10], [11]. Auch für das MedEmergency-Simulationssystem konnte das MedBiquitous-Modell mit einigen Erweiterungen eingesetzt werden. Der Handlungsablauf kann hierbei wie bei anderen Systemen verzweigt und linear vorgegeben sein. Auch problemlösungsorientierte/globale Strukturen mit flexibler Auswahl von Anamnese- und Diagnoseschritten wie beim Web-SP System sind möglich. Zusätzlich kann der Benutzer jederzeit flexibel Therapieoptionen auswählen und die Auswirkungen auf die

„(Vital-)Parameter“ des virtuellen Patienten direkt erfahren und beobachten. Dies wurde durch eine Adaption des MedBiquitous-Standards erreicht, indem die zur Programmsteuerung erforderlichen Counter auch bei Interaktion des Benutzers mit Befund- und Therapieoptionen (VPD-Elementen) des virtuellen Patienten anhand zugehöriger Regeln geprüft werden. Die bisher übliche „Weiterschaltung“ zu einem neuen Handlungsknoten (ActivityNode) durch das Anklicken eines Navigationselements wird umgangen. Die Simulation entspricht so eher einer realen Behandlungssituation als ein festgelegter Ablauf mit sequentiellen Handlungsoptionen.

Bei einem ressourcenschonenden System sollten sich virtuelle Patienten ohne spezielle Fachkenntnisse und möglichst auch für unterschiedliche Fachgebiete unkompliziert erstellen lassen. Vorhandene Systeme zeigen hierbei unterschiedliche Ausprägungen der Anpassbarkeit: von einer für eine Anwendung speziell erstellten Webseite [8] über Systeme für ein einzelnes Fachgebiet mit anpassbaren Inhalten [9] bis hin zu flexiblen Systemen mit flexiblen Inhalten und unterschiedlichen Fachgebieten [4], [6], [7], [10], [11], [12].

Die Anpassungsmöglichkeiten werden z.T. nicht genauer beschrieben, in einigen Fällen finden sich weitergehende Erläuterungen. Das ältere CASE-System verwendet vorgegebene Schablonen (Macromedia Authorware), die ohne Fachkenntnisse ausgefüllt werden können [6]. Die Flexibilität ist hier eingeschränkt, da der Benutzer die Schablonen nicht verändern kann. Ein hohes Maß an Flexibilität bietet die Autorenkomponente des Web-SP Systems. Der Benutzer kann Vorlagen virtueller Patienten mit hinterlegten medizinischen Inhalten benutzen oder existierende Fälle anpassen [10], [11]. Für das MedEmergency-System existiert zurzeit nur eine Player-Komponente, so dass die Inhalte mit einem XML-Editor entsprechend dem MedBiquitous-Standard eingegeben werden müssen. Die XML-Struktur ermöglicht hierbei frei anpassbare Inhalte auch für unterschiedliche Fachgebiete.

Idealerweise sollte sich auch die Software und insbesondere die Benutzeroberfläche leicht anpassen lassen. Dies kann z.B. bei der Verwendung in einem anderen Fachgebiet erforderlich werden. Je spezialisierter ein System ist und umso mehr Inhalte und Programmstruktur verbunden sind, umso aufwändiger ist eine Veränderung. So müsste z.B. eine speziell für einen virtuellen Patienten erstellte Webseite [8] komplett umgeändert bzw. neu erstellt werden.

Das für den MedEmergency-Player eingesetzte PureMVC-Framework ermöglicht hingegen eine strikte Trennung von Datenmodell, Benutzeroberfläche und Programmlogik des Frontends. Die mit Flash Builder erstellte Benutzeroberfläche kann unabhängig vom Rest der Applikation mit geringem Aufwand geändert werden. Die Menüinhalte sind in der XML-Struktur hinterlegt und somit flexibel, so dass ein Einsatz in unterschiedlichen Fachbereichen ohne Änderung der Software erleichtert wird. Auch Variablen, Zähler und Verhalten werden so codiert, dass das Frontend unabhängig vom konkreten virtuellen Patienten ist. Das Frontend erhält die virtuellen Patientendaten unab-

hängig durch Proxies und separate Delegates vom Server. Dadurch werden Änderungen der Serverarchitektur erleichtert.

Für den verbreiteten Einsatz virtueller Patienten sollte sich die Player-Software benutzerseitig unkompliziert und ohne besondere Hardwareanforderungen einsetzen lassen. Hierbei ist nur das älteste System ODSDM/DDx&Tx nicht plattformunabhängig und muss als komplexes System installiert werden, zumal es zusätzliche Komponenten wie einen Videodisk-Player zur Darstellung der virtuellen Patienten verwendet [3], [4], [5]. Der Einsatz für den Benutzer ist somit erschwert. Ältere Systeme auf Basis von Macromedia Authorware werden auf einer CD-ROM als ausführbare Dateien zur Verfügung gestellt [6], [9], wären aber prinzipiell auf Windows- und Mac OS-Systemen ausführbar. Die übrigen Systeme sind bereits webbasiert [7], [8], [10], [11], [12], [13], [14], so dass sie sich für den Benutzer unkompliziert im Webbrowser darstellen lassen. Die Flex-Frontend-Komponente des MedEmergency-Players erlaubt ebenfalls eine plattformunabhängige Darstellung ohne Installationsaufwand im Browser des Anwendercomputers. Durch die abschnittsweise Weitervermittlung bereits vom Server aufbereiteter XML-Daten sind die benutzerseitigen Hardwareanforderungen hierbei gering: Das Frontend muss nicht die gesamten Dateien im Speicher vorhalten und verarbeiten. Zusammenfassend ließ sich mit dem MedEmergency-Player eine anpassbare und plattformunabhängige Software zur Simulation virtueller Patienten für das zahnmedizinische Notfalltraining konzipieren. Hierbei konnte der MedBiquitous-Standard verwendet und adaptiert werden. Zur Steigerung der Realitätsnähe kann der Benutzer im Unterschied zu vorhandenen Systemen die „(Vital-)Parameter“ der virtuellen Patienten direkt beeinflussen und erfahren.

## Ausblick

Der MedEmergency-Player kann auch in anderen (medizinischen) Fachbereichen eingesetzt werden, da (multimediale) Inhalte und Verhalten des virtuellen Patienten in einer XML-Struktur kodiert werden. Auch die Menüinhalte sind dadurch flexibel, wobei die Gliederung in Anamnese, Diagnose und Therapie im Frontend festgelegt ist. Diese ist allerdings bei jeder (zahn-)medizinischen Behandlung relevant. Eine Änderung wird durch das MVC-Entwurfsmuster und die Möglichkeiten von Flash Builder zum vereinfachten Redesign des GUI erleichtert.

Auch innerhalb des Fachgebietes Zahnmedizin sind weitere Einsatzbereiche denkbar, so z.B. die Simulation einer Notfallbehandlung bei akuten Schmerzzuständen. Hierbei ließe sich ein rasches und systematisches Vorgehen mit der MedEmergency-Software trainieren. Eine Anpassung an differierende Lehrmeinungen ist durch Änderung der XML-Inhalte möglich.

Für die Weiterentwicklung des MedEmergency-Players ist die Zukunft der Flash-Plattform von Interesse. Trotz der weiten Verbreitung wird die Bedeutung des Flash-

Players als Webbrowser Plug-In weiter abnehmen, während HTML/JavaScript die entsprechenden Aufgaben übernimmt. Diese Entwicklung ist momentan allerdings noch „im Fluss“, so dass eher für die Flash-Plattform ausgereifte „Komplettlösungen“ zur Softwareentwicklung verfügbar sind. Eine aufwändige Anpassung des MedEmergency-Players an verschiedene Browser(-versionen) ist nicht erforderlich. Die Anwendung hat einen begrenzten Adressatenkreis und lässt sich alternativ auf mobilen und Desktop-Geräten mit der Laufzeitumgebung Adobe Air betreiben. Selbst eine Portierung nach HTML/JavaScript wäre aufgrund des modularen Aufbaus denkbar; die Änderung beträfe zudem nur das Frontend. Die Programmlogik ließe sich u.U. mit einer ausgereiften Version des FlexJS-SDK nach JavaScript übersetzen. Weiterhin ist das verwendete PureMVC-Framework von plattformspezifischen Strukturen wie den Flash Event-Klassen unabhängig und existiert auch in einer Version für JavaScript.

Bevor weitere Funktionen implementiert werden, sollte die Software systematisch evaluiert werden. Der MedEmergency-Player könnte zunächst ergänzend zu einer Ausbildungsmaßnahme oder zum Selbststudium online zur Verfügung gestellt werden. Je nach Zielgruppe können sinnvolle Veränderungen und Erweiterungen erfolgen. Möglich wäre z.B. ein Bewertungssystem oder ein Autorensystem zur einfacheren Erstellung virtueller Patienten.

## Anmerkungen

## Interessenkonflikte

Der Autor erklärt, dass er keine Interessenkonflikte in Zusammenhang mit diesem Artikel hat.

## Danksagung

Mein besonderer Dank gilt Herrn Prof. Dr. Martin Haag, Hochschule Heilbronn für fachlichen Rat und Unterstützung.

## Literatur

- Kononowicz AA, Zary N, Edelbring S, Corral J, Hege I. Virtual patients – what are we talking about? A framework to classify the meanings of the term in healthcare education. *BMC Med Educ.* 2015 Feb;15:11. DOI: 10.1186/s12909-015-0296-3
- Ellaway R, Candler C, Greene P, Smothers V. An Architectural Model for MedBiquitous Virtual Patients. Baltimore: MedBiquitous Consortium; 2006 [cited 2017 Jan 11]. Available from: <http://groups.medbiq.org/medbiq/display/VPWG/MedBiquitous+Virtual+Patient+Architecture>
- Finkelstein MW, Johnson LA, Lilly GE. Interactive videodisc patient simulations of oral diseases. *J Dent Educ.* 1988 Apr;52(4):217-20.
- Clancy JM, Johnson L, Finkelstein MW, Lilly GE. Dental diagnosis and treatment (DDx&Tx): Interactive videodisc patient simulations for dental education. *Comput Methods Programs Biomed.* 1990; 33(1):21-6. DOI: 10.1016/0169-2607(90)90019-6
- Finkelstein MW, Johnson LA, Lilly GE. A computer management system for patient simulations. *Comput Methods Programs Biomed.* 1991;34(4):257-61. DOI: 10.1016/0169-2607(91)90109-7
- Abbey LM, Arnold P, Halunko L, Huneke MB, Lee S. CASE STUDIES for Dentistry: development of a tool to author interactive, multimedia, computer-based patient simulations. *J Dent Educ.* 2003 Dec;67(12):1345-54.
- Schittek Janda M, Mattheos N, Nattestad A, Wagner A, Nebel D, Färböom C, Lê DH, Attström R. Simulation of patient encounters using a virtual patient in periodontology instruction of dental students: design, usability, and learning effect in history-taking skills. *Eur J Dent Educ.* 2004 Aug;8(3):111-9. DOI: 10.1111/j.1600-0579.2004.00339.x
- Boynton JR, Green TG, Johnson LA, Nainar SM, Straffon LH. The virtual child: evaluation of an internet-based pediatric behavior management simulation. *J Dent Educ.* 2007 Sep;71(9):1187-93.
- Littlefield JH, Demps EL, Keiser K, Chatterjee L, Yuan CH, Hargreaves KM. A multimedia patient simulation for teaching and assessing endodontic diagnosis. *J Dent Educ.* 2003 Jun;67(6):669-77.
- Zary N, Johnson G, Fors U. Web-based virtual patients in dentistry: factors influencing the use of cases in the Web-SP system. *Eur J Dent Educ.* 2009 Feb;13(1):2-9. DOI: 10.1111/j.1600-0579.2007.00470.x
- Lund B, Fors U, Sejersen R, Sallnäs EL, Rosén A. Student perception of two different simulation techniques in oral and maxillofacial surgery undergraduate training. *BMC Med Educ.* 2011 Oct;11:82. DOI: 10.1186/1472-6920-11-82
- Clark GT, Suri A, Enciso R. Autonomous virtual patients in dentistry: system accuracy and expert versus novice comparison. *J Dent Educ.* 2012 Oct;76(10):1365-70.
- Papadopoulos L, Pentzou AE, Louloudiadis K, Tsiatsos TK. Design and evaluation of a simulation for pediatric dentistry in virtual worlds. *J Med Internet Res.* 2013 Oct;15(11):e240. DOI: 10.2196/jmir.2651
- Antoniou PE, Athanasopoulou CA, Dafli E, Bamidis PD. Exploring design requirements for repurposing dental virtual patients from the web to second life: a focus group study. *J Med Internet Res.* 2014 Jun;16(6):e151. DOI: 10.2196/jmir.3343
- Cook DA, Triola MM. Virtual patients: a critical literature review and proposed next steps. *Med Educ.* 2009 Apr;43(4):303-11. DOI: 10.1111/j.1365-2923.2008.03286.x
- Triola MM, Champion N, McGee JB, Albright S, Greene P, Smothers V, Ellaway R. An XML standard for virtual patients: exchanging case-based simulations in medical education. *AMIA Annu Symp Proc.* 2007 Oct:741-5.
- MedBiquitous Consortium. Virtual Patient Implementers. [cited 2017 Oct 18]. Available from: [https://www.medbiq.org/virtual\\_patient/implementers](https://www.medbiq.org/virtual_patient/implementers)
- Noaman AY, Al Mansour AA. A Comparative Study Between Two Types of Database Management Systems: XML-Enabled Relational and Native XML. *World Appl Sci J.* 2012; 19(7):972-85.
- Pfeil C. Adobe AIR. RIAs für den Desktop entwickeln. Know-how für HTML/Ajax- und Flash/Flex-Entwickler. München: Addison-Wesley; 2009.

20. Adobe Systems. Adobe's view of Flex and its commitments to Flex in the future. San Jose: Adobe Systems; 2012 [cited 2017 Oct 18]. Available from: <http://www.adobe.com/devnet/flex/whitepapers/roadmap.html>
21. Hall C. ActionScript Developer's Guide to PureMVC. Sebastopol: O'Reilly Media; 2011.
22. Smothers V, Azan B, Ellaway R. ANSI /MEDBIQ VP.10.1-2010 MedBiquitous Virtual Patient Specifications and Description Document. Baltimore: MedBiquitous Consortium; 2010 [cited 2017 October 18]. Available from: [http://www.medbiq.org/working\\_groups/virtual\\_patient/VirtualPatientDataSpecification.pdf](http://www.medbiq.org/working_groups/virtual_patient/VirtualPatientDataSpecification.pdf)
23. Azan B, Smothers V. ANSI /MEDBIQ VP.10.1-2010. MedBiquitous Virtual Patient Player. Specifications and Description Document. Baltimore: MedBiquitous Consortium; 2010 [cited 2017 Oct 18]. Available from: [http://medbiquitous.org/working\\_groups/virtual\\_patient/VirtualPatientPlayerSpecification.pdf](http://medbiquitous.org/working_groups/virtual_patient/VirtualPatientPlayerSpecification.pdf)

**Korrespondenzadresse:**

Dr. Martin Geneit  
Praxis für Zahnmedizin und Oralchirurgie Dr. med. dent.  
Martin Geneit, Ludwigsweg 6, 49152 Bad Essen,  
Deutschland  
geneit@t-online.de

**Bitte zitieren als**

Geneit M. Interaktives Notfalltraining in der Zahnmedizin: Entwicklung einer Virtual Patient Player-Simulationssoftware. *GMS Med Inform Biom Epidemiol.* 2018;14(3):Doc14.  
DOI: 10.3205/mibe000192, URN: urn:nbn:de:0183-mibe0001920

**Artikel online frei zugänglich unter**

<http://www.egms.de/en/journals/mibe/2018-14/mibe000192.shtml>

**Veröffentlicht:** 13.11.2018

**Copyright**

©2018 Geneit. Dieser Artikel ist ein Open-Access-Artikel und steht unter den Lizenzbedingungen der Creative Commons Attribution 4.0 License (Namensnennung). Lizenz-Angaben siehe <http://creativecommons.org/licenses/by/4.0/>.