

# Autorenwerkzeug Javanti

Christian Kohls, Tobias Windbrake

e-Learning Lab  
Fachhochschule Wedel  
Feldstrasse 143  
22880 Wedel  
kls@fh-wedel.de  
wb@fh-wedel.de

**Abstract:** Javanti ist ein Autorenwerkzeug, mit dem sich interaktive Präsentationen und Lernkurse entwickeln lassen. Es dient vor allem der besseren Visualisierung von Informationen und Wissen. Auf virtuellen Folien lassen sich visuelle Elemente platzieren. Die Elemente können statisch, dynamisch oder interaktiv sein. Mit Hilfe einer visuellen Sprache und einer Skriptsprache sind Eigenschaftsveränderungen und Aktionen zur Laufzeit einer Präsentation oder eines Kurses möglich. Zur Strukturierung der Folien wird ein Zeitstrahl verwendet. Animation und verschiedene Aufgabentypen lassen sich ohne Programmierung mit Javanti festlegen.

## 1. e-Learning und Visualisierung

Ein wesentlicher Aspekt rechnerunterstützter Lehr- und Lernformen sind die verbesserten Visualisierungsmöglichkeiten. Computer erlauben die Animation, Manipulation und eine flexible Verwaltung von visuellen Lernmaterialien. Ein Beispiel ist der Einsatz multimedialer Angebote, bei denen Text, schematische Animation, Videoaufnahmen und Tonwiedergabe kombiniert werden. Im Gegensatz zu den traditionellen Medien können alle Codierungsformen in einer Anwendung gebündelt werden, wobei ein direkter, sequenzieller, strukturierter oder verlinkter Zugriff erfolgen kann [1]. Der Rechneinsatz hebt sich dadurch hervor, dass mit den Illustrationsmaterialien interagiert werden kann. Im einfachsten Beispiel können Elemente der visuellen Darstellung verschoben, verändert oder ausgeblendet werden und so zu einer dynamischen Repräsentation führen. Der Interaktionsgrad kann aber auch so weit gehen, dass die visuellen Elemente mit hinreichend Logik ausgestattet sind und sie im Rahmen von Simulationen für virtuelle Experimente eingesetzt werden können. Natürlich ist es auch möglich, im Rahmen des Unterrichts am Rechner ganz neue Inhalte gemeinsam mit den Lerneteilnehmern zu entwickeln. Spezialisierte Werkzeuge wie Mindmapping-Tools sind ein Beispiel für die effiziente Erarbeitung von strukturierten Tafelbildern. Diese Potentiale lassen sich sowohl im Präsenz- als auch im Fernstudium didaktisch sinnvoll ausschöpfen.

Das Entwickeln der Inhalte ist in der Regel jedoch kein trivialer Prozess, denn die erhöhten Interaktionsmöglichkeiten erfordern die Berücksichtigung verschiedener Präsentationszustände. Auch die Produktion hochwertiger Animation ist mit Aufwand verbunden. Die Arbeit erleichtern können dabei Autorensysteme wie das an der Fachhochschule Wedel entwickelte Javanti [2]. Javanti ermöglicht das Erstellen interaktiver Präsentationen und Lernkurse. Ausgestattet mit einem allgemein angelegten Repertoire an Standardelementen und Aktionseinstellungen lassen sich viele der im Lernangebot häufig vorkommenden Interaktionsformen unmittelbar und ohne Programmierung festlegen. Zusätzlich erlaubt eine Skriptsprache die Definition echter Verhaltenslogik für Bildelemente. Zielsetzung von Javanti ist es, Lehrbeauftragten und Ausbildern, die über keinerlei Programmierkenntnisse verfügen, die Erstellung hochwertiger Kursmaterialien am Rechner zu erlauben.

## **2. Architektur und Funktionsweise von Javanti**

Javanti ist mittlerweile Bestandteil der CampusSource-Initiative geworden, ein Projekt, bei dem verschiedene Open-Source-Initiativen deutscher Hochschulen ihre Bildungs-Software über ein gemeinsames Portal anbieten [3]. Zudem finden sich dort Kooperationspartner, um die verschiedenen Aspekte einer vollständigen e-Learning- Architektur miteinander zu verknüpfen.

### **2.1 Kategorisierung von Javanti**

Eine grobe Unterteilung der im e-Learning-Umfeld eingesetzten Werkzeuge und Programme differenziert zwischen Software für die Erstellung der Lerninhalte, die Verwaltung von Lernmodulen sowie Lernerdaten und schließlich der clientseitigen Anwendung zur Wiedergabe des Materials [4]. In jedem Bereich können dabei eine oder mehrere Applikationen beteiligt sein. Für die Erstellung des rohen Lernmaterials kommen häufig Text- oder HTML-Editoren, Grafikprogramme und Videoschnittsoftware zum Einsatz. Die Integration der einzelnen Materialien zu einem Lernkurs wird dann von einem Autorensystem übernommen. Gleichzeitig gibt es Produkte am Markt, die mehrere der oben aufgeführten Kategorien belegen. Viele Lern-Management-Systeme (LMS) kümmern sich nicht nur um die Verwaltung von Kursen und Lernerdaten sondern bieten darüber hinaus rudimentäre Autorenfunktionalität an, so zum Beispiel das Anlegen von Multiple-Choice-Aufgaben oder Lückentexten.

Javanti besetzt die Kategorien Materialerstellung im Sinne eines Autorensystems sowie Materialwiedergabe. Die Verwaltung von Lernkursen wird von Javanti nicht abgedeckt. Kurse und Präsentationen können alternativ aus einem lokalen Verzeichnis, von CD-ROM oder über das Internet gestartet werden. Die grundlegende Arbeitsweise von Javanti entlehnt sich den Paradigmen von Präsentationsprogrammen wie PowerPoint oder Star Impress gleichsam wie denen kommerzieller Autorenwerkzeuge wie zum Beispiel Hypercard, Toolbook oder Director.

## 2.2 Arbeitsmetapher

Die zentrale Arbeitsmetapher von Javanti ist das Erstellen virtueller Folien an einer Tafel. Eine Folie entspricht dabei genau einer Bildschirmseite, auf der sich verschiedene visuelle Elemente platzieren lassen. Es gibt in Javanti eine Auswahl unterschiedlichster Elemente, die sich leicht erweitern lässt. Insbesondere ist es auch dem Anwender möglich, eigene Elementtypen zu entwickeln. Der Basisumfang von Javanti umfasst statische, dynamische und interaktive Elemente. Als statisch werden jene Elemente einer Folie betrachtet, die zur Laufzeit eines Kurses oder einer Präsentation zunächst einmal invariant sind. Dazu gehören Grafiken aus Bilddateien, Texte oder HTML-Seiten. Zudem lassen sich mit Javanti vektorbasierte Grafiken selbst erstellen. Dazu stehen Freihand-, Kreis-, Rechteck-, Vieleck- und Bézierwerkzeuge zur Verfügung. Während der Präsentation können sich diese Elemente nur verändern, wenn Aktionsskripte eine Eigenschaft neu setzen. Bei dynamischen Elementen ist eine Veränderung eines Elements über die Zeit implizit gegeben. Dazu zählen beispielsweise die Video- und Soundwiedergabe oder Animationspfade. Interaktive Elemente sind bereits mit einem Basisverhalten für Benutzereingaben ausgestattet. Hierzu gehören Antwortfelder, Zeichenflächen, Lückentexte, Multiple-Choice-Tests, Anordnungs- und Zuordnungsaufgaben. Es gibt zum Beispiel ein Gruppierungselement, mit dem sich eine Anordnungsaufgabe ohne Programmierung festlegen lässt, indem die einzelnen Kindelemente der Gruppe zur Erstellungszeit in der richtigen Reihenfolge angeordnet werden. Zur Kurslaufzeit werden die Elemente durchgemischt und der Lerner kann diese durch Verschieben neu ordnen. Das Anordnungselement überwacht dabei diesen Vorgang und kann beim Bewerten der Aufgabe ein adäquates Feedback geben.

Jedem Element können außerdem verschiedene Verhaltensstrategien für Benutzereingaben oder Ereignisse zugewiesen werden. So kann festgelegt werden, wie die Präsentation reagiert, wenn der Benutzer auf ein Folienelement mit der Maus klickt oder eine Aufgabe korrekt gelöst hat. Das Verhalten wird entweder mit einer Skriptsprache oder einer visuellen Sprache ausgedrückt. Als Skriptsprache steht ein erweiterter Befehlssatz der Tool Command Language (Tcl) zur Verfügung [5]. Mit Tcl lassen sich die Eigenschaften von Folienelementen setzen oder animieren. Auf diese Weise können auch statische Elemente während der Laufzeit verändert werden. Die visuelle Sprache bietet zudem für die häufigsten Aktionen einen sehr intuitiven Weg, Interaktion zu definieren. Auf sie wird weiter unten näher eingegangen.

Die Bearbeitung der Folien geschieht im WYSIWYG-Modus (What-You-See-Is-What-You-Get). Der Kursentwickler sieht auf seinem Bildschirm die Folie genau so, wie sie später auch der Lernende betrachten wird. Dabei gilt es allerdings zu beachten, dass sich während der Laufzeit einer Präsentation oder eines Lernkurses die Inhalte dynamisch verändern können. Insofern lassen sich zur Erstellungszeit nicht alle möglichen Visualisierungszustände abbilden, sondern nur der jeweilige Ausgangspunkt einer Darstellung.

### **2.3 Zeitstrahl zur Strukturierung der Folien**

Die Strukturierung der Folien in Javanti erfolgt entlang eines Zeitstrahls (Timeline). Der Zeitstrahl definiert in horizontaler Ausrichtung die Anordnung der Folien. Ein Abspielkopf, der entlang des Zeitstrahls zu diskreten Positionen (Seiten) bewegt werden kann, bestimmt, welche Folie aktuell angezeigt wird. Entlang des Zeitstrahls lassen sich mehrere Folienebenen übereinander ausrichten. Dadurch ist es möglich, mehrere Folien übereinander zu legen, gleichzeitig auf dem Bildschirm anzuzeigen, und somit deren Inhalte zu kombinieren. Aus jeder Folienebene wird genau eine Folie angezeigt, nämlich die Folie, die sich gerade unterhalb des Abspielkopfes befindet. Eine Folie innerhalb der Ebene kann dabei mehrere aufeinanderfolgende diskrete Positionen auf dem Zeitstrahl belegen und so für mehrere Bildschirmseiten gültig sein. Dies ist sinnvoll, wenn man mehrere Folienebenen kombiniert, wie ein Beispiel verdeutlicht. Auf oberster Ebene können Folien platziert werden, deren Anzeigedauer jeweils genau einer Position auf dem Zeitstrahl entspricht. Mit jeder Neuplatzierung des Abspielkopfes ändert sich der dargestellte Inhalt auf dem Bildschirm, es handelt sich also um Inhaltsfolien. Eine weitere Folienebene beinhaltet dagegen nur eine einzige Folie, deren Anzeigedauer sich über den gesamten Zeitstrahl erstreckt. Da sich der Abspielkopf stets innerhalb dieser Folie befinden wird, ändern sich deren Inhalte niemals auf dem Bildschirm. Diese Folie kann also für invariante Bestandteile eines Kurses wie Hintergrundgrafiken oder Navigationsstrukturen verwendet werden. Eine dritte Folienebene kann für Kapitelüberschriften oder kapitelspezifische Elemente verwendet werden. Die Folien dieser Ebene erstrecken sich entlang des Zeitstrahls jeweils über die Länge eines Kapitels. Die Inhalte der Hintergrundebene ändern sich also niemals, die der Kapitelebene ändern sich nach jedem Kapitel, und auf der Inhaltsebene findet auf jeder Seite eine Veränderung statt. Da beliebig viele Ebenen erlaubt sind, lassen sich Präsentationen und Kurse in angemessener Granularität gliedern.

### **2.3 Keyframebasierte Animation**

Ein weiteres Merkmal von Javanti ist die Definition von keyframebasierten Zeichentricksequenzen. Wenn in zwei aufeinanderfolgenden Folien einer Ebene die gleichen Elemente vorkommen, können die Folien als Schlüsselszenen einer Animation interpretiert werden. Dazu lässt sich für das Element auf der ersten Folie die Animation einschalten. Als Konsequenz werden die Eigenschaftswerte des ersten Elementes schrittweise in Richtung der Werte des Elementes der zweiten Folie interpoliert. Erfolgt die Annäherung schnell genug, führt dies zu einer animierten Darstellung.

### 3. Interaktion und Javanti

Die Verwendung bewegter Bilder ist bereits ein großer Visualisierungsgewinn [6]. Im Gegensatz zur gedruckten Wiedergabe, bei der nur Momentaufnahmen berücksichtigt werden können, wird der zeitlichen Dimension Rechnung getragen. Dargestellte Einzelprozesse können den realen Verlauf besser repräsentieren, da sie Parallelität oder die korrekte Ablaufsequenz berücksichtigen. Das Zusammenspiel der einzelnen Elemente kann besser ausgedrückt werden.

#### 3.1 Nichtdeterministische Animation

Ein weiterer Vorteil des Rechnereinsatzes ist, dass Zeichentricksequenzen nicht deterministisch ablaufen müssen, d.h. der Benutzer kann in den Animationsablauf aktiv eingreifen. Statt der automatischen Animation kann der Anwender Elemente am Bildschirm bewegen, über Schaltflächen oder bestimmte Ereignisse die Animation triggern und/oder beeinflussen. Auch für Videosequenzen lässt sich prinzipiell durch Nonlinearität die Determiniertheit innerhalb gewisser Grenzen aufbrechen, wenn die einzelnen Filmabschnitte sinnvoll unterteilt und quer miteinander verlinkt werden können.

Javanti erlaubt den unmittelbaren Eingriff in das Animationsgeschehen sowohl für Präsenz- als auch Selbststudiums Anwendungen. Die einfachste Methode, während einer Präsentation bewegte Veränderungen darzustellen, ist das Verschieben, Vergrößern oder Rotieren von Elementen. Für jedes Element kann einzeln eingestellt werden, ob es auf diese Weise direkt vom Benutzer verändert werden darf. Voraussetzung für den Einsatz ist dabei das Wissen des Präsentierenden, welche Veränderungen auf der Folie semantisch sinnvoll sind. Wenn man einen Kurs für das Selbststudium konzipiert, erfordert nichtdeterministische Animation stets die Definition von Verhaltenslogik für die betroffenen Elemente. Die Entwicklung ist also etwas aufwändiger, da das „Wissen“ des Präsentierenden über die Bedeutung der Elemente einer Folie softwaretechnisch nachempfunden werden muss.

Auch bei Präsenzpräsentationen ist die Definition von Verhaltenslogik für Elemente sinnvoll. So könnte zum Beispiel das Pflanzen eines Samens in den Erdboden zum Wachsen eines Baumes führen. In Javanti hieße das:

Wenn das Samenelement (also ein Bildelement, das einen Samen darstellt) über das Bodenelement gezogen wird, dann mache das Baumelement sichtbar und vergrößere es.

Die Verhaltenslogik besteht hier aus einer Kollisionserkennung für zwei Elemente und einer hierfür festgelegten Eigenschaftsveränderung des dritten Elementes. Beides lässt sich über die visuelle Sprache von Javanti ohne Eingabe von Programmcode abbilden.

### 3.2 Interaktive Folien

Weil der Benutzer während der Präsentationszeit aktiv in das Geschehen eingreifen kann, verwendet Javanti hierfür die Bezeichnung *interaktive Folie*. Der Interaktionsgrad einer Präsentation oder eines Kurses kann dabei stark variieren; dies gilt für e-Learning-Anwendungen allgemein. Ausschlaggebend für eine Bewertung ist dabei die tatsächliche Vielfalt der Eingabe- und Reaktionsmöglichkeiten. Auch klassische Präsentationssoftware ist interaktiv, weil der Anwender über die Folien navigieren kann. Der Interaktionsgrad ist jedoch ausgesprochen gering und kann vernachlässigt werden. Standardtestformen wie Multiple-Choice, Anordnungs- oder Zuordnungsaufgaben weisen einen höheren Interaktionsgrad aus, die Zahl der Eingabe- und Feedbackmöglichkeiten bleibt aber begrenzt. Den höchsten Interaktionsgrad erreichen Anwendungen, bei denen der Benutzer eine freie Wahl der Eingabeparameter hat (zum Beispiel für Experimente) und aktiv das Geschehen beeinflussen kann. Die Reaktion einer virtuellen Folie auf Benutzereingaben ist im letzteren Falle nicht mehr trivial und erfordert eine programmierte Logik. Im Rahmen der Präsenzlehre kann oft auf vordefinierte Verhaltensweisen verzichtet werden, da der Präsentierende selbst die angemessenen Veränderungen auf der Folie vornehmen kann.

### 3.3 Skalierbarkeit der Einsatzgebiete

In der Regel muss davon ausgegangen werden, dass der Anwender eines domänenunabhängigen Autorenwerkzeuges über keinerlei oder nur sehr geringe Programmierkenntnisse verfügt. Aus diesem Grunde unterstützt Javanti sehr häufig vorkommende Interaktionsprozesse mit vorgefertigten Komponenten. Damit lassen sich zwar nicht alle Problemstellungen abdecken, aber aus genau diesem Grunde gibt es auch noch eine integrierte Skriptsprache sowie eine Schnittstelle zu Java-Programmen. Insofern ist Javanti sehr skalierbar und spricht Anwender mit unterschiedlichen Vorkenntnissen an. Statische Folien lassen sich wie mit anderen Präsentationsprogrammen ebenfalls leicht erstellen. Zusätzlich lassen sich interaktive Elemente, wie beispielsweise Übungsaufgaben, auf den Folien platzieren. Für einfache Aktionen – Verändern von Elementeigenschaften oder Ausführen von Operationen – steht eine visuelle Sprache zur Verfügung. Komplexere Aktionen und Programmlogik lassen sich über Tcl abbilden. Durch die Definition neuer Elementtypen in Java lassen sich Folienbestandteile auch in einer objektorientierten Sprache abbilden.

### 3.4 Interaktionsdefinition durch eine visuelle Sprache

Die visuelle Sprache von Javanti ist ein Kompromiss zwischen Ausdrucksstärke der Sprache und einfacher Handhabbarkeit für den Laien. Obgleich die Sprache als Visuelle Programmiersprache (VPS) aufgefasst werden kann, ist sie nicht dafür konzipiert worden, echte Programmstrukturen abzubilden [7]. Visualisiert werden einerseits die Parameter einer Aktion, zum anderen lassen sich Regeln an ein oder mehrere Elemente einer Folie binden. Die Sprache lässt sich somit am ehesten als beispielorientierte und regelorientierte VPS klassifizieren [8].

Zunächst lassen sich neben den obligatorischen Ereignissen einer Folie (Mouse-Events) neue Ereignisse visuell modellieren. Aus der Überlagerung mehrerer Elemente lässt sich ein Kollisions- oder Contains-Ereignis ableiten. Wenn die beteiligten Elemente später zur Laufzeit der Präsentation wieder in der gleichen Konstellation auftreten, dann wird die zum Ereignis gehörende Aktionsliste abgearbeitet. Das Gleiche gilt für Anordnungsereignisse, bei denen eine definierte Anordnung mehrerer Elemente zum Auslösen eines Ereignisses führt.

Die Aktionen der Aktionsliste sind wiederum auf einer Folie visuell editierbar. Zu jeder Aktion gehören ein Sender, der die Aktion triggert, und ein Empfänger, der von der Aktion verändert wird. Sender und Empfänger werden durch einen Pfeil miteinander verbunden. Schließlich werden die Auswirkungen der Aktion visualisiert. Führt eine Aktion beispielsweise zu einer Positionsänderung des Empfängerelementes, dann wird auf der Folie auch die neue Position des Elementes durch eine schematische Darstellung des Originalelements dargestellt. Die Zielposition kann dabei auf der Folie intuitiv festgelegt werden und erfordert keine numerische Eingabe über die Tastatur.

Indem man Verhaltensstrategien an Elemente bindet, können sich letztere automatisch über den Bildschirm bewegen. Bewegt man etwa Elemente, denen das Verhalten „Container“ zugewiesen wurde, dann werden alle Elemente mitgezogen, die sich über dem Containerelement befinden. Magnetisches Verhalten führt dazu, dass Folien-elemente, die sich in der Nähe des magnetisierten Elementes befinden, angezogen werden. Der reziproke Fall ist das Schubverhalten eines Elementes, bei dem alle kollidierenden Elemente zur Seite geschoben werden. Definiert man dieses Verhalten etwa für ein verschiebbares Bildelement, das einen Bulldozer zeigt, so kann man zur Laufzeit der Präsentation mit diesem Element Dinge auf dem Bildschirm zusammenschieben.



Die visuelle Sprache umfasst eine Reihe weiterer graphischer Ereignisse, Aktionen und Verhaltensregeln, mit denen sich interaktive Verhaltensweisen für Folien ohne Programmierung festlegen lassen.

Constraintbasierte Verhaltensdefinition

## **4. Einsatzgebiete**

Folien lassen sich mit Hilfe der in Javanti verfügbaren Sprachen schrittweise aufbauen, indem die Elemente nacheinander eingeblendet werden. Die Anzeigereihenfolge muss dabei nicht im Voraus festgelegt sein. Mit Hilfe der ereignisgetriggerten und animierten Eigenschaftsveränderung lassen sich Teilschritte eines Prozesses deutlicher darstellen. Die einzelnen Phasen können dabei schrittweise durchlaufen werden - interessant beispielsweise für Wachstumsprozesse in der Natur. In Einzelschritten lassen sich Folienelemente entfernen oder hinzufügen. Auch der Bewegungsprozess von Elementen kann in einzelne Abschnitte unterteilt werden. Zu Illustrationen lassen sich beschreibende Texte und Hintergrundinformationen dynamisch ein- und ausblenden. Die Beschriftung der Folien kann sich kontextbezogen verändern, etwa in Abhängigkeit von der Zielgruppe oder bei wiederholter Anzeige. Grafiken lassen sich vergrößern und verkleinern, so dass zwischen unterschiedlichen Detaillierungsgraden gewechselt werden kann. Durch die temporäre Veränderung von Eigenschaftswerten lassen sich Elemente fokussieren, etwa kann der gerade erläuterte Vortragspunkt einer Folie rot eingefärbt werden. Zudem können Elemente vollständig in ihrem Erscheinungsbild verändert werden, so dass auch Metamorphoseprozesse – die Wandlung einer Raupe zum Schmetterling – dargestellt werden können.

### **4.1 Einsatz in der Präsenzlehre**

In der Präsenzlehre lässt sich Javanti als guter Ersatz für Tafel und Overheadprojektor einsetzen. Das dynamische Erzeugen von Elementen zur Laufzeit einer Präsentation erlaubt auch die spontane und dynamische Entwicklung neuer Tafelbilder in Kooperation mit der Lernergruppe. Es lassen sich zudem domänenspezifische Elementbibliotheken anlegen, die das Arbeiten mit Javanti im Unterricht sehr effizient gestalten. In der Chemie lässt sich so auf ein Repertoire an verschiedenen chemischen Stoffen zurückgreifen, Physiker können Grundschaltungen einfach auf die virtuelle Tafel ziehen, Informatiker greifen auf Basiselemente für Klassendiagramme zurück. Jede Folie lässt sich um Notizen und Freihandzeichnungen ergänzen. Als Eingabemedien kommen nicht nur Tastatur und Maus in Frage, auch stiftbasierte Eingabegeräte erweisen sich als besonders sinnvoll. Hierzu gehören Grafiktablets, Tablet-PCs oder interaktive Tafeln. Vor allem beim Einsatz mit interaktiven Tafeln bildet Javanti einen vollständigen Ersatz für traditionelle Tafeln, jedoch erweitert um die beschriebenen Visualisierungs- und Interaktionsmöglichkeiten. Da sich Javanti-Präsentationen auch auf mehreren Rechnern synchronisieren lassen, kann ein gruppenorientiertes Arbeiten im Rechenzentrum oder Computerraum stattfinden. Das Tafelbild kann dabei mit den Einzelplatzrechnern in Einklang gebracht werden. Das Ebenenkonzept des Zeitstrahls wird hierbei genutzt, um eine differenzierte Synchronisation zu erreichen. Es kann für jede einzelne Ebene eingestellt werden, ob diese mit allen Rechnern (Public Layer), mit den Rechnern einer Arbeitsgruppe (Group Layer) oder gar nicht (Private Layer) synchronisiert wird.



## 4.2 Einsatz im Selbststudium

Aufgrund der in Javanti integrierten Aufgabentypen, den Strukturierungsmöglichkeiten der Timeline und der Programmierbarkeit lassen sich Kurse auch sehr gut für das Selbststudium konzipieren. Ein interessanter Aspekt dabei ist, dass Präsentationen für die Präsenzlehre um zusätzliche Folien oder Ebenen mit detaillierterem Text oder Übungsaufgaben ergänzt werden können. Die gleiche Kursdatei kann so für Präsenz- und Fernlehre verwendet werden. Autoren müssen sich zudem nur in eine einzige Software einarbeiten, um Inhalte für beide Lehr-/Lernformen anbieten zu können.

Für den Lerner im Selbststudium bietet Javanti die Möglichkeit, Lesezeichen zu setzen, die Inhalte zu restrukturieren und auf den Folien Notizen oder Zeichnungen einzufügen. Die Folien können sequentiell, hierarchisch oder über Direktzugriff angesteuert werden. Eine Vernetzung der Folien über Hyperlinks ist ebenfalls möglich. Synchronisiert man auf entfernt vernetzten Rechnern einen Javanti-Kurs, dann kann kooperativ mit den Inhalten gearbeitet werden. Eine spontane Lernsitzung kann dabei jederzeit initiiert werden, da kein zusätzlicher Server für die Kommunikation der Javanti-Kurse benötigt wird.

## 5. Ausblick

Javanti wurde über einen Zeitraum von drei Jahren entwickelt und hat den erfolgreichen Einsatz in der Praxis gezeigt. Doch auch Javanti kann den erhöhten Entwicklungsaufwand für interaktive Lernangebote nicht aus der Welt schaffen, sondern nur reduzieren. Die erweiterten Visualisierungsmöglichkeiten werden schließlich auch durch ein größeres Angebot an Bild- und Videomaterial erreicht, deren Produktion zusätzlich anfällt. Illustrationen, die in gedruckten Medien schon aus Platzgründen auf eine Darstellung reduziert werden, können bei computergestützter Lehre in mehrere Grafiken aufgeteilt werden. Dies führt zu quantitativ mehr Medienelementen, und die Einzelteile müssen bedeutsam miteinander verknüpft werden. Sinnvoll erscheint es daher, den Austausch von bereits erstellten Kursen und Kursbestandteilen über das Internet zu fördern. Gleiches gilt für den Austausch von neuen Elementtypen, die zum Beispiel physikalische Simulationen oder mathematische Graphen darstellen können. Es ist nicht sinnvoll, dass jeder Anwender seine eigenen Erweiterungen für Javanti entwickelt und so das Rad mehrfach erfunden wird. Andererseits soll die Standarddistribution von Javanti nicht mit hochspezialisierten Elementen überladen werden. Hier gilt es, ein Konzept zu entwickeln, das den Austausch neuer Plugins fördert und auch eine Strategie bereithält, bei einer Installation fehlende Programmteile aus dem Netz nachzuladen.

Ein weiteres Forschungsgebiet für die Zukunft ist die Evaluierung der visuellen Sprache im Praxiseinsatz. Das Konzept der Sprache ist aus den empirischen Erfahrungen der Javanti-User-Gemeinde entstanden. Ob die Sprache im Alltagstest tatsächlich eine Vereinfachung der Interaktionsdefinition ermöglicht, vor allem aber, ob sie auch für Nicht-Programmierer verwendbar ist, muss noch untersucht werden. Dabei gilt es insbesondere zu beobachten, mit welcher Strategie die Endanwender an den Produktionsprozess herangehen.

## Literaturverzeichnis

- [1] Seufert, S. / Back, A. / Häusler, M (2001). E-Learning - Weiterbildung im Internet. Smart Books Publishing
- [2] <http://www.javanti.org>
- [3] <http://www.campussource.de>
- [4] Brückner, T. / Schuster, F.: Companion - eine integrierte Lern- und Autorenumgebung für den virtuellen Hochschulverbund Karlsruhe (ViKar),  
<http://zemm.ira.uka.de/~brueckner/documents/papers/CompanionLearntec2000.pdf>
- [5] <http://www.tcl.tk>
- [6] Schmann, H.; Müller, W. (1999): Visualisierung. Grundlagen und allgemeine Methoden, Springer Verlag
- [7] Chang, S.K.: Introduction to Visual Languages and Visual Programming,  
<http://www.cs.pitt.edu/~chang/365/365sy1.html>
- [8] Schiffer, S. (1998): Visuelle Programmierung