

CPoint – ein invasiver, semantischer Editor für wiederverwendbaren Content in MS PowerPoint

Andrea Kohlhasse

School of Engineering and Computer Science / FB Mathematik und Informatik, AG
DiMeB (Digitale Medien in der Bildung)
International University Bremen / Universität Bremen
Campus Ring 1 / Bibliotheksstr. 1
28759 Bremen / 28359 Bremen
a.kohlhasse@iu-bremen.de

Abstract: CPoint („Content for PowerPoint“) ist ein semantischer Editor für E-Learning Content in Form von MS PowerPoint-Objekten. Er ist als zusätzliches Menü innerhalb des Präsentationseditors MS PowerPoint (PPT) realisiert. Diese invasive Implementierung einer Open Source Software innerhalb einer proprietären Applikation basiert zum einen auf einer user-zentrierten Herangehensweise, die die Bekanntheit und breite Anwenderschaft von PPT nutzt, zum anderen auf der Idee, PPT-Präsentationen – eigentlich ein Werkzeug zur visuellen Optimierung von Wissenstransfer – auch als Wissensdatenbank zu begreifen und durch semantische Annotierung verwaltbar zu machen.

1 Einführung

An vielen Hochschulen und wissenschaftlichen Einrichtungen wird inzwischen Präsentationssoftware wie OpenOffice Impress oder MS PowerPoint (PPT) für die Lehre eingesetzt. Eine solche Software ermöglicht die visuelle und strukturelle Optimierung des Wissenstransfers von Lehrendem zu Lernenden im Instruktionsstil. Insbesondere existieren an vielen Hochschulen gemeinsame Wissenspools zur Vorlesungsunterstützung in Form von über die Jahre angesammelten PPT-Präsentationen - so auch an der Carnegie Mellon University, Pittsburgh. Hier wurde innerhalb des Projekts CCaps (Course Capsules, 2002 - 2005) die Idee geboren, diesen Wissenspool leichter zugänglich und wiederverwendbar zu machen. Insbesondere wurde beabsichtigt, das darin enthaltene Wissen dann als Lernobjekte für Studenten (und spätere Alumni) zeit- und ortsunabhängig über das World Wide Web (WWW) in einem "Lebenslanges-Lernen"-Programm (z.B. in Form von E-Lectures) zu nutzen. Die Umsetzung dieser Idee mündete in dem von der Autorin implementierten Softwaresystem „**CPoint**“ – **Content**

for **PowerPoint**¹, dessen Architektur und weiterentwickelte Funktionalitäten in diesem Papier im Folgenden motiviert und näher beschrieben werden sollen. Zum Schluss wird als Ausblick die Nutzung von CPoint in konkreten Lernszenarios angedacht.

2 CPoints Designprinzipien

Der Widerspruch zwischen dem abstrakten Potential und der konkreten Nutzung von Software im Bildungskontext (in [KK04] als „User Riddle“ diskutiert) ist besonders eklatant, wenn von den Benutzern *zusätzlicher* Aufwand zur *späteren* Nutzung des Potentials verlangt wird. Der Designstandpunkt, von dem aus CPoint spezifiziert wurde, ist **subjektorientiert**: nur wenn die Benutzung von Software vom Subjekt aus gesehen Sinn macht, d.h. wenn subjektive Handlungsbegründungen existieren (siehe [Hol95]), wird sie angenommen. Aus dieser „Mikrosicht“ muss der Benutzer als Mensch mit seiner je eigenen Vergangenheit, Gegenwart und Zukunft verstanden werden. Das „User Riddle“ erklärt sich dann aus der typischerweise einseitigen Ausrichtung von Softwaredesign auf die Zukunft (siehe [Ko05]).

Die erste fundamentale Designentscheidung für CPoint bestand nun darin, die Vergangenheit des Benutzers mit in den Blick zu nehmen und die angedachte, unterstützende Software als sogenannten „**invasiven Editor**“ innerhalb von MS PowerPoint zu implementieren: die anvisierte CPoint-Software sollte dem Benutzer nicht nur denselben Look-and-Feel der schon für ihn bekannten Präsentationssoftware bieten, sie sollte auch aus PowerPoint heraus aufrufbar und bedienbar sein (siehe Abb. 1). Die Motivation des Einsatzes von invasiver Software besteht darin, dass sie den Aufwand und/oder die Scheu eines Benutzers beim Benutzen neuer Softwarekomponenten reduziert, gleichzeitig aber die Stärke von Open Source Anwendungen auch innerhalb proprietärer Applikationen ausgenutzt werden kann². Daher wurde CPoint in der mit MS PowerPoint ausgelieferten Programmierumgebung basierend auf der Programmiersprache „Visual Basic for Applications“ als **PPT-Add-In** implementiert.



Abbildung 1: Die CPoint-Menüleiste

Die zweite wesentliche Entscheidung beim CPoint-Design wurde bezüglich des Modularitätslevels getroffen, d.h. es musste entschieden werden, welche Granularität die Lernobjekte erhalten sollten. Da die Wiederverwendbarkeit solcher Lerneinheiten stark kontextabhängig ist und sich Kontexte - insbesondere im schulischen Bereich - immer

¹ Veröffentlicht ist es unter der Gnu Lesser General Public License (LGPL) auf <http://kwarc.eecs.u-bremen.de/projects/CPoint/install.html>.

² Eine ausführliche Beschreibung findet sich in [KK04] und [Ko05].

wieder neu stellen, bietet es sich an, die Lernobjekte als „natürliche kleinste Einheiten“ (im Sinne von **Microcontent**, von A. Leene in [Le06] charakterisiert durch die Eigenschaften „focused“, „self-contained“, „indivisible“ und „addressable“) zu konzipieren. Dieser feine Granularitätslevel wurde umgesetzt, indem mit CPoint PPT-**Objekte** wie Textboxen oder Bilder als Wissensbausteine annotiert und verwaltet werden können.

Weiterhin sollte möglichst viel Inhalt „eingefangen“ werden können. Nun versucht ein PPT-Autor bei der Herstellung einer PPT-Präsentation sowohl sein explizites (z.B. Fakten) als auch implizites Wissen (z.B. Struktur) darzustellen. Präsentationssoftware wird gerade deswegen gerne zur Wissensvermittlung eingesetzt, weil sie das explizite Darstellen von implizitem Wissen, z.B. durch einen visuellen Aufbau von Strukturen, erleichtert. Im „Flow“ eines Vortrags verbinden sich dann (hoffentlich) alle Komponenten zu einem Ganzen. Um nun viele der darin enthaltenen Inhalte „einfangen“ zu können, muss ein PPT-Objekt mehr Informationen als den reinen Objekthinhalte erhalten – es muss semantisch annotiert werden. Deswegen wurde CPoint als **semantischer Editor** von PPT-Microcontent konzipiert. Die durch diese semantische Annotation angehefteten Zusatzinformationen dürfen selbstverständlich die Präsentationseigenschaften einer PPT-Show nicht stören. Daraus ergibt sich die Frage, wo die zusätzlichen Daten denn gespeichert werden sollen bzw. können. Um eine einfache Wiederverwendbarkeit zu garantieren, werden diese Daten nun von CPoint nicht nur in den entsprechenden PPT-Präsentationen, sondern sogar in der **Datenstruktur der PPT-Objekte selbst** gespeichert. So gehen z.B. bei einem „Copy-und-Paste“-Vorgang die annotierten Informationen nicht verloren³.

Um nun solch einen semantischen Microcontent über das WWW auswertbar zu machen, muss es in einem Format vorliegen können, das zum einen webfähig ist und zum anderen von Applikationen „verstanden“ werden kann. Leider geht aber durch den Formalisierungsgrad einer großen Maschinenverständlichkeit die Menschenverständlichkeit verloren, so dass das gesuchte Format nicht zu formal sein sollte. Als sinnvollen Mittelweg bietet sich hier das pragmatische semantische XML-Format **OMDoc** (Open Mathematical Documents [Ko06]) an, auf das auch schon existierende E-Learning-Applikationen - wie beispielsweise das ActiveMath System [MA+01] - aufbauen⁴. Die vom Content-Autor zu leistende Annotation seiner PPT-Objekte ist dementsprechend auf das Zielformat OMDoc gerichtet, so dass PPT-Lernobjekte mittels CPoint „leicht“ in OMDoc-Objekte konvertiert werden können.

Im Folgenden soll nun genauer auf die verschiedenen CPoint-Funktionalitäten eingegangen werden, die aus einer subjektorientierten „Mikrosicht“ die Gegenwart des Benutzers betreffen und das aktuelle Handeln bzw. Annehmen der Software mitbewirken.

³ Allerdings verlieren die Objekte ihren (räumlichen) Kontext, wodurch ihre Referenzen zum Teil ungültig werden.

⁴ Obwohl OMDoc logisch-mathematisch orientiert ist, versteht es sich durchaus als allgemeines E-Science Content-Format.

3 CPoint im Detail

Bevor die semantischen Hauptfunktionalitäten von CPoint erklärt werden, wollen wir uns zunächst typische *Inhalte einer PPT-Folie* anschauen. In der Abbildung 2 sehen wir eine PPT-Folie, die aus einem Titelobjekt, zwei umrahmten Textboxen und einer labelartigen Textbox besteht. Der Text selbst stellt nur zum Teil das Wissen dar, das vermittelt werden soll. Zum Beispiel ordnet der Titel die kommende Information in einen Kontext ein: Es geht hier um das Konzept der Äquivalenzrelation. Weiterhin handelt es sich bei dem Text in der größeren Textbox offensichtlich um eine Definition des Begriffs "Äquivalenzrelation" und bei der kleineren, gerahmten Textbox um einfache Beispiele für Äquivalenzrelationen. Letzteres wird suggeriert durch den räumlichen Zusammenhang zwischen dem Text "Beispiele" und der Textbox. Dieses **implizite Wissen** ist wesentlich für die Inhalte dieser PPT-Folie und soll von Content-Autoren mittels CPoint "eingefangen" werden können.

Das Diagramm zeigt eine PPT-Folie mit dem Titel "Äquivalenzrelation". Die Folie ist in drei Hauptbereiche unterteilt:

- Titel:** Äquivalenzrelation
- Definition:** Die Relation " \sim " ist eine **Äquivalenzrelation**, wenn (für alle a, b und c)

$a \sim a$	reflexiv
$a \sim b \text{ gdw } b \sim a$	symmetrisch
$a \sim b \text{ und } b \sim c \Rightarrow a \sim c$	transitiv
- Beispiele:**

▪ " $=$ "	reflexiv, symmetrisch, transitiv
▪ " zusammenhängend "	reflexiv, symmetrisch, transitiv

Die Folie ist mit vier Gedankenbläschen versehen, die implizites Wissen darstellen:

- Oben links: "Die Existenz eines 'äquivalent zu'- Objekts wird vorausgesetzt."
- Oben rechts: "Um dieses Konzept geht es."
- Mitte links: "Dieser Text definiert eine Äquivalenzrelation."
- Unten links: "Das Wort 'Beispiele' bezieht sich auf den darunterliegenden Kasten."

Die Folie ist mit der Nummer 4 unten rechts beschriftet.

Abbildung 2: Implizites Wissen einer PPT-Folie (in CPoints Hide-Modus)

3.1 Vom impliziten zum expliziten Wissen: CPoints Hauptfunktionalitäten

Beim **Kategorisieren** kann der Inhalt eines PPT-Objekts (aus einer vorgegebenen Liste von Kategorien, die sich am OMDoc-Format als Systemontologie orientiert) klassifiziert werden. So kann die Textbox mit den einfachen Beispielen für eine

Äquivalenzrelation als "Beispiel" kategorisiert werden und den Namen "Einfache, unmittelbar einleuchtende Äquivalenzrelationen" zugewiesen bekommen. In Abbildung 3 visualisiert CPoint diese Information durch automatisch erzeugte Label, die die vom Content-Autor für die entsprechenden Objekte erzeugten Annotationen der Kategorie und des Namens enthalten.

Durch den persönlichen Akt der Namensgebung werden den Objekten auch persönliche Bedeutungen zugewiesen, an die sich später dann leichter erinnert werden kann. Dies wird in CPoints **Navigationshilfe** „GoTo“ ausgenutzt. Das Navigieren (Finden und Springen) ist nicht nur innerhalb einer Präsentation möglich, sondern auch über ganze Gruppen von anderen PPT-Shows hinweg. Außerdem kann durch die zugrundeliegende Semantik der PPT-Objekte auf vielen Ebenen (z.B. nach Beispielen, deren Namen die Zeichenkette "Äquivalenzrelation" in einem gewissen Vorlesungszyklus enthalten) gesucht werden.

Äquivalenzrelation

- Die Relation " \sim " ist eine **Äquivalenzrelation**, wenn (für alle **a**, **b** und **c**)

$a \sim a$	reflexiv
$a \sim b \text{ gdw } b \sim a$	symmetrisch
$a \sim b \text{ und } b \sim c \Rightarrow a \sim c$	transitiv

Beispiele

- " $=$ " reflexiv, symmetrisch, transitiv
- "**zusammenhängend**" reflexiv, symmetrisch, transitiv

Theory1 (text)
 „Äquivalenzrelationen“

Definition2 (text)
 „Bedingte Äquivalenzrelation“
äquivalentZu

Example4 (text)
 „Einfache, unmittelbar einleuchtende Äquivalenzrelationen“

4

Abbildung 3: Explizites Wissen einer PPT-Folie (in CPoints Visualize-Modus)

Auf der Folie in Abbildungen 2 und 3 ist auch ersichtlich, dass es sich bei der Beispiel-Textbox um Beispiele für "Äquivalenzrelationen" handelt. Diese Beziehung kann vom Content-Autor über CPoints **Referenziersystem** expliziert werden. Jeder Kategorie stehen dabei gewisse Referenziermöglichkeiten durch CPoint zur Verfügung. So ist beispielsweise diese Definition einer Äquivalenzrelation nur sinnvoll innerhalb einer "Äquivalenzen"-Theorie. Die Arbeit im CCaps-Projekt hat gezeigt, dass eine solche Kontextzuweisung unumgänglich ist, beispielsweise kann es sehr unterschiedliche Definitionen für "Bäume" innerhalb der Biologie (z.B. Apfelbaum) oder der Informatik (z.B. Knoten und Kanten) geben.

3.2 Mehrwert-Dienste

CPoints Architektur und Funktionalitäten erklären sich aus den Softwareanforderungen. Darüberhinaus werden dem Anwender mit CPoint aber noch zusätzliche Dienste (Mehrwert-Dienste) angeboten, die zur Nutzung weiter motivieren bzw. anreizen sollen.

Zum einen bietet CPoint einfache Mehrwert-Dienste bei der Entwicklung von PPT-Folien an, z.B. den „**Chopper**“, der Listen in Textboxen in jeweils kleinere Textboxen mit demselben Layout zer„schlägt“ und den „**Shaper**“, mit dem mehrere Textboxen in eine Textbox zusammengefasst werden können. Durch ein technisches User-Interface „**CMath**“ wird die Erstellung von technischen Texten mit vielen Symbolen erleichtert⁵. Zum anderen werden Mehrwert-Dienste in Form von *Zusatzmodulen* angeboten, die unabhängig voneinander geladen werden können.

Nutzern sollte eine möglichst einfache Handhabung von Software zur Verfügung gestellt werden. Da sich Arbeitsabläufe stark an den jeweiligen Funktionen bzw. Rollen des Benutzers orientieren, kann auch die Handhabung auf solche Rollen hin optimiert werden. Im Modul „**CPointAuthor**“ wurde ein Panel (statt einer Menüleiste) realisiert, in der jeweils die semantischen Hauptdaten eines selektierten PPT-Objekts angezeigt und/oder kreierte werden können (zur Veranschaulichung siehe das vereinfachte, deutsche "CPointBasic"-Panel in der folgenden Abbildung). Zusätzlich können im CPointAuthor-Panel auf Knopfdruck kategorisierte Objekte generiert werden, deren Layout durch ein vorher angegebenes CSS-Stylesheet bestimmt wird. Aufbauend auf diese CSS-Datei kann mit dem Modul „**CPointImport**“ auch ein OMDoc-Dokument nach PPT importiert werden, welches aber noch einiger liebevoller Zuwendung bedarf, um ein akzeptables Layout zu erhalten.

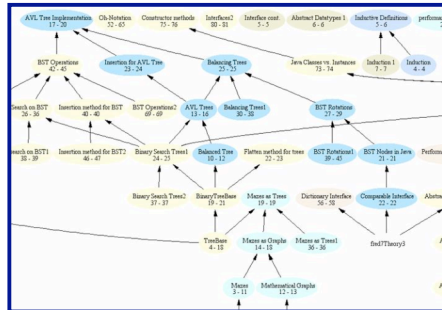


zur Auswahl).

Um einen „cognitive overload“ eines (neuen) Benutzers zu verhindern, wurde die ursprünglich sehr komplexe CPoint-Menüleiste im Modul „**CPointBasic**“ auf wesentliche Funktionalitäten reduziert und mit einem vereinfachten CPointAuthor-Panel (siehe linke Abbildung) kombiniert. Die Idee besteht darin, dass, wenn sich für einen Benutzer eine gewisse Vertrautheit mit CPointBasic einstellt, er sich im Folgenden schrittweise neue Module aneignen kann. Weiterhin ist CPointBasic prinzipiell sprachunabhängig programmiert worden (zur Zeit stehen Deutsch und Englisch

Ein weiteres CPoint-Modul namens „**CPointNotes**“ dient zur strukturierten Kommentierung der PPT-Objekte. Es können Kommentargruppen gebildet werden, wie beispielsweise eine Gruppe mit dem Titel „Noch unzufrieden“ oder „Fehlende Referenz“, welche dann jeweils unabhängig voneinander navigiert werden können.

⁵ Es handelt sich hier um eine starke Erweiterung von Neculas „Texpoint“-Tool (siehe [Ne03]), das die Eingabe von LaTeX-Text mit anschließender PPT-Transformation erlaubt.



Das Modul „CPointGraphs“ verschafft einem Benutzer einen Überblick über die zugrundeliegende semantische Struktur, d.h. die Abhängigkeiten der Objekte voneinander unter einem speziellen Blickwinkel. Dies kann auf verschiedenen Ebenen geschehen, die Ausführlichste ist die Graphdarstellung von Theorie-Objekten (d.h. Konzepten) und ihren Import-Relationen, z.B. innerhalb eines Kurses wie sie in

nebenstehender Abbildung zu sehen ist. Unterschiedliche Knotenfarben repräsentieren dabei verschiedene PPT-Präsentationen. In der Abbildung sieht man einen Ausschnitt eines Informatik-Grundkurses, der an der Carnegie-Mellon-University gegeben wurde. Interessant ist die Beobachtung, dass durch die semantische Annotation des Kurses und der graphischen Aufbereitung sich beim Autor das mentale Modell des Kurses stark veränderte. Beispielsweise führten isoliert dastehende Knoten im Theoriegraphen zu einer Reflektion des Vorlesungsaufbaus.

4 Zusammenfassung und Ausblick

Im vorliegenden Papier wurde der semantische, invasive Editor „CPoint“ zur effizienten Erstellung von E-Learning Content, auf Basis von (zur unterstützenden Wissensvermittlung) in der Lehre erzeugten PPT-Objekten, motiviert und vorgestellt. Bei der Konzeption spielte das Problem des größeren Zeit- und Energieaufwands eines Nutzers für einen zukünftigen Mehrwert eine entscheidende Rolle. Die Effizienz wird durch ein subjektorientiertes Softwaredesign erreicht, das die Historizität eines Nutzers im Blick hat. Die Umsetzung dieser *Mikrosicht* erlaubt den Einsatz von CPoint als Werkzeug zum persönlichen Wissensmanagement.

Die ursprüngliche Idee, einen Wissenspool (in Form von PPT-Präsentationen) mit CPoint zugänglicher und wiederverwendbarer zu machen, ist bisher noch nicht realisiert, d.h. das *konkrete* Potential von Kollaboration im (im weitesten Sinn) Semantic Web über den Austausch von PPT-Objekten ist eine zukünftige Aufgabe. Konkret werden daher zur Zeit Weiterentwicklungen von CPoint aus der *Makrosicht* vorangetrieben:

Einerseits soll die E-Learning Komponente „CPointStudent“ entstehen, mit der Wissensbausteine eines (lokalen) Wissenspools vom Lernenden zusammengestellt werden können und so eine intensive und individuelle Beschäftigung mit dem dargebotenen Wissen (und dadurch ein intensiverer Lerneffekt) erzielt werden kann.

Andererseits arbeitet die Autorin an einem Konzept des konkreten Einsatzes von CPoint im schulischen Kontext (Arbeitstitel: „**Gestuftes Blended Learning and Knowing**“), das davon ausgeht, dass Einsatz von Software in Stufen erfolgen sollte, in denen sich zum einen private Bereiche zu öffentlichen hin öffnen, zum anderen die Rollen eines Lehrenden und eines Lernenden mit der Zeit auflösen. Allerdings muss im Sinne einer

Community of Practice zunächst eine Entwicklung vom Newcomer zum Experten stattfinden, bevor die Rollendichotomie aufgehoben werden kann. In der „letzten“ Stufe verschwinden dann die Grenzen im Sinne des Web2.0 (siehe z.B. Michael Kerres in [Ke06]). Insbesondere werden in diesem Konzept die Komponenten Wissensmanagement und E-Learning dynamisch miteinander verbunden.

Innerhalb des OnceCS-Projekts [On05] soll CPoint als Editor für E-Learning Content im OMDoc-Format eingesetzt werden, der in das Wissenssystem „Connexions“ [Con] eingespeist wird. Zusätzlich wird an der International University Bremen versucht, CPoint von MS PowerPoint auf MS Word zu migrieren („CWord“), um einen weiteren invasiven, semantischen Editor zur effizienten Erstellung von E-Learning Content zu erhalten.

Literaturverzeichnis

- [Con] Connexions, <http://cnx.org/>
- [Hol95] Holzkamp, K.: Lernen: Subjektwissenschaftliche Grundlegung. Campus Verlag, 1995.
- [Ke06] Kerres, M.: Web2.0 and its implications for eLearning. In Proceedings of Microlearning 2006, Innsbruck, Austria. 2006.
- [KK04] Kohlhase, A.; Kohlhase, M.: CPoint: Dissolving the Author's Dilemma. In (Asperti, A., Bancerek, G., Trybulec, A. Hrsg): Mathematical Knowledge Management, MKM'04, number 3119 in LNAI. Springer Verlag, 2004.
- [Ko06] Kohlhase, M.: OMDoc – Open Mathematical Documents. LNAI-Band (im Druck), Springer Verlag, 2006.
- [Ko05] Kohlhase, A.: Overcoming Proprietary Hurdles: CPoint as Invasive Editor. In: Proceedings of Open Source for Education in Europe: Research and Practise, Open University of the Netherlands, Herleen. CD-ROM, 2005, S. 51-56.
- [Le06] Leene, A.: Microcontent is Everywhere. In: Proceedings of Microlearning 2006, Innsbruck, Austria. 2006.
- [MA+01] Melis, E., Andres, E., Franke, A., Goguadse, G., Libbrecht, P., Pollet, M., Ullrich, C.: (2001). ActiveMath System Description. In (Moore, J.D.; Luckhard Redfield C., Johnson W. L. Hrsg), *Artificial Intelligence in Education*, 68. IOS Press, 2001 S. 580 – 582.
- [Ne03] Necula, G.: Texpoint 2.0.3. Online Documentation. Viewed on 2005/10/07 at <http://raw.cs.berkeley.edu/texpoint/TexPoint.html>, 2003.
- [On05] Once-CS: Open Network of Centres of Excellence in Complex Systems, Homepage <http://complexsystems.lri.fr/Portal/>, 2005.