

# Graphbasierte Modelle als Kollaborationsmedien

Niels Pinkwart

HCI Institute  
Carnegie Mellon University  
5000 Forbes Avenue  
Pittsburgh, PA 15213, USA  
niels@pinkwart.de

**Abstract:** Dieser Artikel beschreibt einen theoretischen Ansatz und ein darauf aufbauendes Softwaresystem zur Unterstützung von Kollaboration mittels graphbasierter Modelle - die Arbeit verbindet Aspekte interaktiver Systeme (in Echtzeit manipulierbare und simulierfähige Modelle) mit einer flexiblen Definition von Kooperationszenarien. Zunächst werden in diesem Artikel die Begriffe visueller getypter Graphen und "Reference Frames" als formale Notationen für kollaborativ nutzbare Modelle bzw. interoperable Modellierungssprachen eingeführt. Der Artikel stellt Cool Modes, eine Implementierung des Reference Frame-Ansatzes auf Basis einer flexiblen Softwarearchitektur, vor. Diese Anwendung verwaltet mehrere extern definierbare Reference Frames und unterstützt Kollaboration mit Hilfe gemeinsam nutzbarer Arbeitsbereiche. Hierbei werden Syntaxregeln, Modellinterpretations- und Simulationsmöglichkeiten ebenso unterstützt wie partielle Modellsynchronisation unter Berücksichtigung semantischer Konsistenz: über spezielle Benutzerschnittstellen zu "Reference Frames" wird es Anwendern ermöglicht, heterogene Modelle (die aus Elementen verschiedener Sprachen zusammengesetzt sind) gemeinsam zu erstellen und zu nutzen. Abschließend beschreibt dieser Artikel einige Anwendungsbeispiele aus dem Lehr/Lernbereich, in denen Cool Modes im Rahmen von EU-Forschungsprojekten und Unterrichtsreihen an Schulen eingesetzt worden ist.

## 1 Einleitung

Eine der wesentlichen Funktionen in vielen modernen Softwaresystemen besteht darin, Benutzern interaktive Umgebungen zur Verfügung zu stellen, in denen sie Daten modellieren, analysieren und manipulieren können, sowie Simulationen auf Basis dieser Daten durchführen können. Hier gibt es ein breites Spektrum von Anwendungen in verschiedensten beruflichen Bereichen (z.B. spezialisierte Simulationstools oder Workflowsysteme) wie auch im Gebiet der Lehr/Lernsoftware. Im letzteren wird interaktiven Modellierungsumgebungen ("mind tools") die Eigenschaft zugeschrieben, "intellektuelle Partner" für Lernende darzustellen und diese zum kritischen Denken anzuregen. [Jon00].

Mit zunehmender Vernetzung von Computern in den letzten Jahrzehnten besteht eine weitere essentielle Funktion von Computersystemen darin, Benutzer in deren Kommunikation

und Interaktion zu unterstützen bzw. diese zu ermöglichen. In vielen Bereichen des Lebens nutzen wir heute digitale Technologien wie e-Mail, Internettelefonie oder Webforen, um mit anderen zu kommunizieren. Auch "Zusammenarbeit" (als speziellere Form der Kommunikation) wird durch eine Reihe von Systemen zum computergestützten kollaborativen Lernen und Arbeiten (CSCW/CSCL) ermöglicht. Synchrone Zusammenarbeit wird oft z.B. durch "Shared Workspace"-Systeme ermöglicht, asynchrone Zusammenarbeit oft durch gemeinsam nutzbare Dokumentenarchive.

Die Synthese von interaktiven Systemen zur Modellierungs- und Simulationsunterstützung mit modernen Kommunikationstechnologien führt zu gemeinsam verwendbaren Systemen, die sowohl die Interaktion mit dem Modell unterstützen als auch deren Verwendung als Medium für Zusammenarbeit - z.B. in der Form, dass ein Modell gemeinsam entwickelt und diskutiert wird, wobei Simulationsläufe nahtlos in den Prozess integriert werden können. Systeme dieser Art wurden insbesondere im Bereich des computergestützten Lernens oft gefordert und als "kollaborative Mindtools" bezeichnet [Hop04], sind aber bisher noch nicht in einer Form verfügbar, die Freiheit hinsichtlich der Wahl der Modellierungssprache(n) mit Flexibilität in der Definition des Kollaborationsprozesses verbindet.

In diesem Artikel liegt der Schwerpunkt auf graphbasierten Repräsentationen als Basis für Modellierungs- und Simulationsanwendungen. Diese eignen sich aus mehreren Gründen besonders für einen Einsatz als "kollaborative Mindtools": die Graphstruktur vereinfacht eine partielle Modellsynchronisation (jeder Nutzer arbeitet an "seinem" Teil des Modells, und nur ein Subgraph wird gemeinsam verwendet), Graphen haben sich als effektives Kollaborationsmedium erwiesen [SH03], und viele Modellierungssprachen (z.B. UML, Petrietze, Concept Maps) basieren auf graphbasierte Repräsentationen.

Es gibt eine ganze Reihe von Programmen, die Modellierung mit graphbasierten Repräsentationen unterstützen. Teilweise wird, wie z.B. in Visio<sup>1</sup>, auch die integrierte Nutzung von verschiedenen Modellierungssprachen unterstützt. Einige professionelle Metamodellierungssysteme wie z.B. GME [LMB<sup>+</sup>01] erlauben es, eigene Modellierungssprachen mit operationaler Semantik zu definieren. Synchrone Kollaboration wird jedoch in den existierenden professionell orientierten Modellierungssystemen nicht explizit unterstützt, sieht man von der Möglichkeit ab, die Programme per NetMeeting zu nutzen, welches die Möglichkeiten für Kooperationsszenarien eingeschränkt. Die Daten liegen nur auf einem Rechner, eine spezielle Definition von Arbeitsgruppen ist unmöglich, Awarenessunterstützung fehlt ebenso wie die Möglichkeit, nur Teilmodelle zu synchronisieren.

Im Bereich der forschungsorientierten Software, insbesondere im Gebiet des computerunterstützten kollaborativen Lernens und Arbeitens, existieren eine Reihe von Systemen, die Kollaboration mittels graphbasierter Modelle unterstützen. Prominente Beispiele sind z.B. Belvedere [SH03], eine Umgebung zur Unterstützung von Diskurs im Kontext wissenschaftlicher Argumentation, und Co-Lab [vJdJL<sup>+</sup>05], ein integriertes System zur Förderung von kollaborativen Lernprozessen in den Naturwissenschaften. Systeme, die synchrone Kooperation mittels graphbasierter Modelle ermöglichen, sind oft eingeschränkt hinsichtlich ihrer Erweiterbarkeit (Möglichkeit, eigene Sprachen zu definieren und im System zu nutzen), Interoperabilität (integrierte Verwendbarkeit mehrerer Modellierungsspra-

<sup>1</sup><http://office.microsoft.com/en-us/default.aspx>

chen) und/oder der Ausdrucksmächtigkeit der unterstützten Modellierungssprachen (z.B. Fähigkeit zur Modellsimulation).

In den nächsten Abschnitten dieses Artikels wird zunächst ein theoretischer Rahmen zur Beschreibung interoperable und kollaborativ nutzbare graphbasierte Modellierungssprachen gesetzt. Dieser dient als Basis für ein Softwaresystem, welches sowohl flexible Mechanismen zur synchronen Gruppenarbeit bietet als auch semantisch reichhaltige und integriert nutzbare Modelle unterstützt.

## 2 Reference Frames als Darstellung interoperabler und kollaborativer Modellierungssprachen

Drei theoretische Ansätze zur Beschreibung von interoperablen und kollaborativ nutzbaren graphbasierten Modellierungssprachen bieten sich an: Graphentheorie, Metamodellierung und die Theorie visueller Sprachen. Jeder dieser Ansätze hat dabei gewisse Vorzüge: so liefert die mathematische Graphentheorie ein solides Fundament zur Beschreibung von Struktur und algorithmischen Aspekten von Graphen, Metamodellierungsansätze wie MOF<sup>2</sup> sind gut dazu geeignet, Modellinteroperabilität wie auch -transformationen zu beschreiben. Theoretische Beschreibungen visueller Sprachen [MMW98] letztlich erlauben es, graphisch orientierte Sprachen und deren Beziehung exakt zu beschreiben. Keine dieser Theorien reicht jedoch für sich genommen aus, um interaktive Graphen als Kollaborationsmedien angemessen zu beschreiben.

Der hier gewählte Ansatz basiert auf der mathematischen Graphentheorie und fügt Typinformationen für Knoten und Kanten hinzu, um die Verschiedenartigkeit von Elementen in Modellierungssprachen zu berücksichtigen. Weiterhin werden, wie in verschiedenen visuellen Sprachtheorien, in der Notation graphische Attribute (Layoutinformationen etc.) explizit vorgesehen: ein visueller getypter Graph  $\langle G, \mathcal{N}, \mathcal{E}, L \rangle$  besteht aus einem Graph  $G$ , einer Menge von Knotentypen  $\mathcal{N}$ , einer Menge von Kantentypen  $\mathcal{E}$  ( $G$  umfasst eine Abbildung seiner Knoten- und Kantenmenge auf die Typmengen) und einer Layoutabbildung  $L$ , die die visuellen Parameter von Knoten und Kanten typabhängig repräsentiert. Visuelle getypte Graphen sind grundsätzlich dazu geeignet, heterogene graphbasierte Modelle (bestehend aus Elementen mehrerer Modellierungssprachen) zu beschreiben.

Modellierungssprachen beinhalten typischerweise Regeln über syntaktische Korrektheit von Modellen sowie eine Modellsemantik, die oft (insbesondere bei simulierfähigen Modellen) operationalen Charakter hat. Syntaxregeln werden im hier beschriebenen Ansatz durch Integritätsbedingungen ausgedrückt: diese entsprechen Abbildungen von der Menge der visuellen getypten Graphen nach  $\{\text{wahr}, \text{falsch}\}$ . Modellsemantik wird durch ein Tupel  $(D, Ip)$  repräsentiert. Hierbei ist  $D$  die semantische Domäne und  $Ip$  eine Abbildung, die syntaktisch korrekten visuellen getypten Graphen einen Wert aus  $D$  zuweist.  $Ip$  kann hierbei auch auf Basis von Knoten- bzw. Kantensemantik definiert sein.

Ein naheliegendes Ziel bei einer kooperativen Nutzung von Modellgraphen ist es, eine par-

<sup>2</sup><http://www.omg.org/technology/documents/formal/mof.htm>

tielle Synchronisation zuzulassen, in der Benutzer verschiedene Teile eines Modells bearbeiten. Eine solch flexible kollaborative Nutzung von semantikbehafteten Modellgraphen bringt jedoch Probleme hinsichtlich semantischer Konsistenz mit sich, wenn nur Teile eines Modells synchronisiert werden. Dies wird in den beiden in Abbildung 1 gezeigten Petrinetzen deutlich. Diese sind fast strukturgleich und unterscheiden sich nur in einer Stelle und einer Transition. Dieser Unterschied führt jedoch dazu, dass sich die Modellsemantik auch in den gemeinsam vorhandenen Teilen der beiden Modelle unterscheidet. Damit sind die in beiden Arbeitsbereichen vorhandenen Teile des Petrinetzes als Kooperationsmedium denkbar ungeeignet, denn die Konsistenz der gemeinsam verwendeten Daten steht hier im Konflikt mit der semantischen Korrektheit des Modells. Beides ist jedoch für sinnvolle Kooperations Szenarien notwendig: es macht keinen Sinn, von "gemeinsamen" Modellteilen zu sprechen, wenn die lokal verwendeten Daten voneinander abweichen. Andererseits ist aber auch eine Verletzung der formalen Semantik (hier: Aktivierungsbedingungen in Petrinetzen) keine vertretbare Lösung.

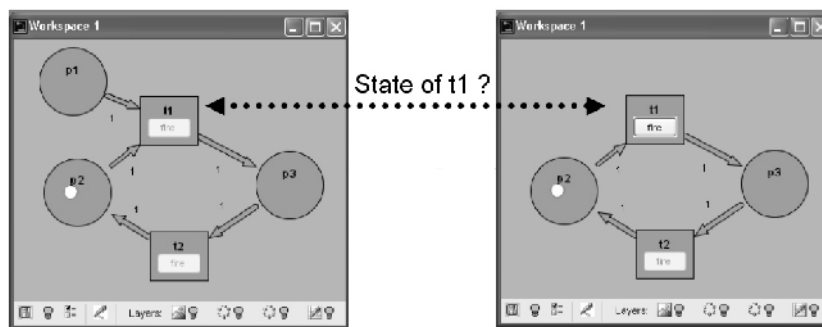


Abbildung 1: Probleme bei teilweiser Modellsynchronisation

Zur Ermöglichung partieller Modellsynchronisation unter Sicherung semantischer Konsistenz (d.h. alle gemeinsam verwendeten Teile des Modellgraphen haben in allen Anwendungen die gleiche lokal berechenbare Semantik) verwendet der hier vorgeschlagene Ansatz *Synchronisationskontexte*. Diese beschreiben, welche Teile eines Gesamtmodells einen Einfluss auf die Semantik einzelner Elemente haben - dadurch wird es Anwendungen ermöglicht, semantische Abhängigkeiten zu lokalisieren und ggf. bei den zur Verfügung gestellten Kollaborationsmöglichkeiten zu berücksichtigen. Bei einer gegebenen Semantikabbildung  $Ip$  ist ein Synchronisationskontext eines Knotens  $n$  in einem visuellen getypten Graphen  $G$   $Sync(n_G)$  ein  $n$  enthaltender Subgraph von  $G$ , der die Semantik von  $n$  erhält:  $Ip(n_G) = Ip(n_{Sync(n_G)})$ . Es kann gezeigt werden, dass minimale Synchronisationskontexte stets existieren, jedoch im Allgemeinen nicht eindeutig sind.

Während visuelle getypte Graphen das formale Pendant zu Modellen darstellen, sind Modellierungssprachen als "Reference Frames" in der hier skizzierten Theorie repräsentiert und bilden deren Kern. Ein Reference Frame  $\mathcal{R} = \langle \mathcal{N}, \mathcal{E}, V_N, V_E, C, D, Ip, S \rangle$  fasst Typinformation ( $\mathcal{N}, \mathcal{E}$ ) und visuelle Attribute ( $V_N, V_E$ ) für Elemente und Relationen einer Modellierungssprache zusammen und spezifiziert zusätzlich eine Menge  $C$  von syntakti-

schen Integriätsbedingungen, eine Semantik  $(D, Ip)$  und eine allgemeine Synchronisationskontextabbildung  $\mathcal{S} : N \times G \mapsto \mathcal{P}(G)$ , welche für einen gegebenen Knoten  $n$  und einen Graphen  $G$  (der  $n$  enthält) einen Subgraph von  $G$  bestimmt, der einen Synchronisationskontext von  $n$  in  $G$  darstellt.

In obiger Definition beschreiben Reference Frames kollaborativ nutzbare visuelle graphbasierte Modellierungssprachen. Interoperabilität zwischen Reference Frames wird in der Theorie durch verschiedene Konzepte, die in [Pin05] näher beschrieben sind, erreicht. So kann ein Import von Elementtypen zwischen Reference Frames ebenso ausgedrückt werden (und somit Graphenelemente logisch zu mehreren Sprachen “gehören”) wie eine Spezialisierung von Reference Frames im Sinne einer “is-a” Relation, wobei verschiedene Optionen für Spezialisierungen (z.B. Erhaltung von Syntaxregeln oder Semantik) in der Reference Frame-Notation ausgedrückt werden können.

Reference Frame-basierte Modellinterpretation ist eine Herausforderung sowohl auf theoretischer Ebene als auch im Systementwurf: ohne einen geeigneten Ansatz zur Interpretation sind aktive, simulierfähige Modelle nicht möglich. Hierbei sind zwei Fragen zentral: wie kann zunächst die Semantik eines heterogenen visuellen getypten Graphs  $G$  durch einen Reference Frame  $\mathcal{R}$  bestimmt werden, wenn nicht alle in  $G$  auftretenden Typen in  $\mathcal{R}$  bekannt sind? Darauf aufbauend stellt sich die Frage nach der Bedeutung von Modellsemantik, wenn mehrere Reference Frames dasselbe Modell interpretieren. Die Beantwortung beider Fragen ist notwendig, will man heterogene Modelle und die integrierte Nutzung mehrerer Reference Frames (beides bringt enormes Anwendungspotenzial mit sich) zulassen. Der in [Pin05] detailliert dargestellte Ansatz basiert auf zwei Prinzipien, um diese Probleme anzugehen.

Für einen Reference Frame  $\mathcal{R}$  werden zunächst die von ihm definierten und importierten Elementtypen (Knoten und Kanten) als  $\text{KNOWS}(\mathcal{R})$  bezeichnet. Für einen visuellen getypten Graph  $\langle G, \mathcal{N}, \mathcal{E}, L \rangle$  bezeichnet dann  $G|_{\mathcal{R}}$  den Subgraph von  $G$ , der nur aus Typen in  $\text{KNOWS}(\mathcal{R})$  besteht. Zerlegt man nun die Interpretationsfunktion von  $\mathcal{R}$  in einen allgemeinen Teil  $Ip_{gen}$  (Berechnung unabhängig von Typinformation möglich) und einen speziellen (domänenabhängigen) Teil  $Ip_{dom}$ , so kann man die durch  $\mathcal{R}$  bestimmbare Semantik wie folgt notieren:

$$Ip(\mathcal{R})(\langle G, \mathcal{N}, \mathcal{E}, L \rangle) = Ip_{gen}(\mathcal{R})(\langle G, \_, \_, L \rangle) \otimes Ip_{dom}(\mathcal{R})(\langle G|_{\mathcal{R}}, \mathcal{N}, \mathcal{E}, L \rangle)$$

Wird nun der visuelle getypte Graph durch eine Menge  $R$  von Reference Frames interpretiert, so lässt sich dies durch eine Kombination von Einzelinterpretationen ausdrücken. Dies ergibt die folgende Beziehung:

$$Ip(\langle G, \mathcal{N}, \mathcal{E}, L \rangle) = \bigotimes_{\mathcal{R} \in R} Ip(\mathcal{R})(\langle G, \mathcal{N}, \mathcal{E}, L \rangle)$$

Die Kombination dieser Gleichungen liefert einen Ausdruck für die Semantik eines visuellen getypten Graphs auf Basis der einzelnen Semantikabbildungen von mehreren Reference Frames:

$$Ip(\langle G, \mathcal{N}, \mathcal{E}, L \rangle) = \bigotimes_{\mathcal{R} \in R} (Ip_{gen}(\mathcal{R})(\langle G, \_, \_, L \rangle) \otimes Ip_{dom}(\mathcal{R})(\langle G|_{\mathcal{R}}, \mathcal{N}, \mathcal{E}, L \rangle))$$

Der hier nicht näher ausgeführte  $\otimes$ -Operator entspricht einer abstrakten Aggregation. Mehrere Alternativen zur konkreten Umsetzung (integrative und separierende Ansätze) sind in [Pin05] detaillierter diskutiert.

### 3 Das System “Cool Modes”

Der im vorigen Abschnitt beschriebene “Reference Frame”-Ansatz zur Beschreibung kollaborativ nutzbarer interoperabler Modellierungssprachen ist zunächst unabhängig von konkreten Implementierungen zu sehen - es macht Sinn, die abstrakte (und vielfältig verwendbare) Theorie von konkreten Softwareumsetzungen zu trennen. In diesem Abschnitt wird Cool Modes (COllaborative Open Learning and MODEling System), eine Implementierung des Reference Frame-Ansatzes, kurz vorgestellt. Cool Modes ist auf den Bereich der Lehr/Lernanwendungen zugeschnitten und unterstützt Kollaboration mit Hilfe gemeinsam nutzbarer Arbeitsbereiche.

#### 3.1 Systemarchitektur

Technisch ist Cool Modes mittels einer Schichtenarchitektur realisiert. Hierbei kommen in den unteren Schichten die Bibliotheken MatchMaker [Jan03] zur Datenreplikation und Kommunikationsunterstützung sowie der JGraph<sup>3</sup>, eine Klassenbibliothek zur Darstellung von Graphen, zum Einsatz. Eine eigene Architekturschicht für abstrakte Reference Frames erlaubt es, Modellierungssprachen systemextern und unabhängig von einer konkreten Benutzerschnittstelle in Java oder XML zu spezifizieren - auf dieser Schicht wird Modellinterpretation und -operabilität ermöglicht sowie die Konsistenz von Modellsemantik unter partieller Kopplung (Synchronisationskontexte) sichergestellt. Die oberste Schicht in der Architektur stellt schließlich die Benutzerschnittstellen zur Verfügung und ermöglicht es, Modellierungssprachen (Reference Frames) zur Laufzeit zum System hinzuzufügen und integriert in mehreren Arbeitsbereichen zu verwenden. Hierbei lassen sich die Arbeitsbereiche auf verschiedene Arten kooperativ nutzen, die von einer kompletten Anwendungssynchronisation über die Kopplung einzelner Arbeitsbereiche bis hin zur Kooperation mittels einzelner Elemente (mitsamt deren Synchronisationskontext) reichen.

Abbildung 2 zeigt die Benutzerschnittstelle von Cool Modes und illustriert die Visualisierung der partiellen Synchronisation. Auf der rechten Seite des Screenshots ist eine “Palette” (die Ansicht eines Reference Frames) zu sehen - hier mit Elementen zur Konstruktion von Additionsnetzen. Rechts oben sind die Symbole dreier weiterer Paletten (Handschrift, mathematische Graphen, System Dynamics) zu sehen. Mit Hilfe dieser Buttons kann der Benutzer die angezeigte Palette wechseln. Elemente der aktuell angezeigten Palette lassen sich jederzeit zu den Modellen in den Arbeitsbereichen per Drag&Drop hinzufügen.

Die Abbildung zeigt drei Arbeitsbereiche, die in unterschiedlicher Form synchronisiert

<sup>3</sup>[http://www.collide.info/howtos/jgraph\\_howto](http://www.collide.info/howtos/jgraph_howto)

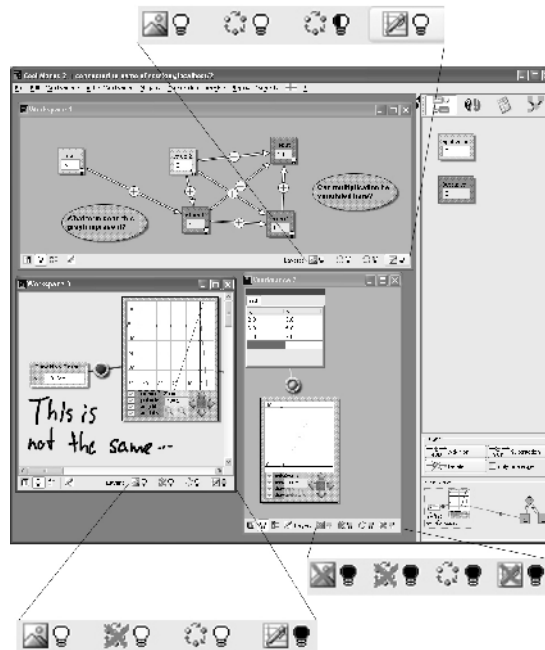


Abbildung 2: Partielle Synchronisation in Cool Modes

sind. Dies wird jeweils durch die Metapher einer Glühbirne symbolisiert. Drei der Ebenen von Arbeitsbereich 1 (oben) sind vollständig gekoppelt. Die Modellebene ist nur teilweise synchronisiert: das Modell ist eine Mischung aus einem Rechnernetz und Kommentarknoten, wobei nur das Rechnernetz den Kooperationspartnern gezeigt wird. Arbeitsbereich 2 (rechts unten) enthält eine Datentabelle und deren grafische Visualisierung. Dieser Arbeitsbereich ist privat (alle Glühbirnen sind dunkel). Arbeitsbereich 3 (links unten) hat drei synchronisierte Ebenen, nur die Handschriftebene ist privat. Die Annotation "This is not the same" zu den beiden vom Funktionsplotter angezeigten Graphen ist somit nicht für die Kooperationspartner sichtbar. Arbeitsbereich 3 ist größer als dargestellt: ganz rechts unten zeigt ein Übersichtsfenster, welches auch zur Darstellung von Awareness-Information dient, den gesamten Inhalt an.

### 3.2 Einsatzgebiete

Die Entwicklung des Cool Modes Systems wurde im Jahr 2001 begonnen, seitdem wird die Umgebung kontinuierlich weiterentwickelt. Im Rahmen verschiedener forschungs- und anwendungsorientierter Projekte wurden bis heute mehr als 50 verschiedene teils standardisierte, teils zielgerichtet erschaffene grafische Modellierungssprachen als "Re-

ference Frames” beschrieben und mit Cool Modes bzw. mit der verwandten FreeStyler-Anwendung [Gaß03], die von Cool Modes durch eine seitenorientierte Benutzerschnittstelle abweicht, verwendet. Innerhalb des letzten Jahres wurde Cool Modes mehr als 700mal von der “offiziellen” Webseite heruntergeladen, die Downloads der einzelnen Modellierungssprachen summieren sich auf mehr als 2000.

Ein konkretes Beispiel für ein Forschungsprojekt, in dem Cool Modes erfolgreich eingesetzt wurde, ist COLDEX<sup>4</sup>. In diesem von 2001 bis 2005 laufenden EU-Projekt war es das Ziel, interkulturelle Lernergruppen mittels Softwaresystemen, die verteiltes kollaboratives Lernen im Bereich der Naturwissenschaften unterstützen, zu fördern. Der in COLDEX propagierte pädagogische Ansatz eines “herausforderungsbasierten Lernens” sah Szenarien vor, in denen Lernende mit realistischen und komplexen wissenschaftlichen Fragen konfrontiert wurden. “Digitale Experimentier-Toolkits” (DEXTs) beinhalteten digitale und reale Werkzeuge, die es den Lernenden erlaubten, Nachforschungen anzustellen. In einigen DEXTs kam Cool Modes (mit verschiedensten Modellierungssprachen) zum Einsatz. Die Ausdrucksmächtigkeit der Reference Frames-Implementierung (volle Java-Mächtigkeit) war hierbei ein wichtiger Erfolgsfaktor. Sie erlaubte es, Cool Modes durch die Modellierungssprachen mit realen physikalischen Geräten zu verbinden und Daten auszutauschen. Drei Beispiele hierfür:

**Astronomie.** In der Astronomie-Anwendung können durch Teleskope aufgenommene Mondfotos in Cool Modes verwendet und nachbearbeitet werden. Mittels einfacher geometrischer Regeln können die Höhen von Mondkratern im System durch Rechenetze bestimmt werden.

**Seismik.** In der Erdbebenanwendung ist es das Ziel, Epizentren und Hypozentren von Erdbeben mit Hilfe von authentischen Sensordaten zu berechnen. Der Berechnungsprozess wird dabei durch eine speziell entworfene Modellierungssprache unterstützt.

**Labyrinth.** Im Labyrinth-Szenario (siehe Abbildung 3) sollen Lernende einem Roboter aus einem Labyrinth helfen, indem sie ihm Aktionen in Form von einfachen “wenn-dann” Entscheidungsregeln, die bei Bedarf spezifiziert werden, beibringen. Das entsprechende DEXT beinhaltet sowohl eine Softwareanwendung (auf Basis von Cool Modes) als auch ein reales Labyrinth, in dem ein Lego Mindstorms-Roboter agiert. Ein Datentransfer von Entscheidungsregeln zwischen diesen Umgebungen ist möglich.

## 4 Evaluation und Ausblick

Zahlreiche erfolgreiche Praxiseinsätze von Cool Modes in der Schul- und Hochschullehre belegen die Eigenschaft dieses Systems als “kollaboratives Mindtool”, das lernförderlich

---

<sup>4</sup><http://www.coldex.info>



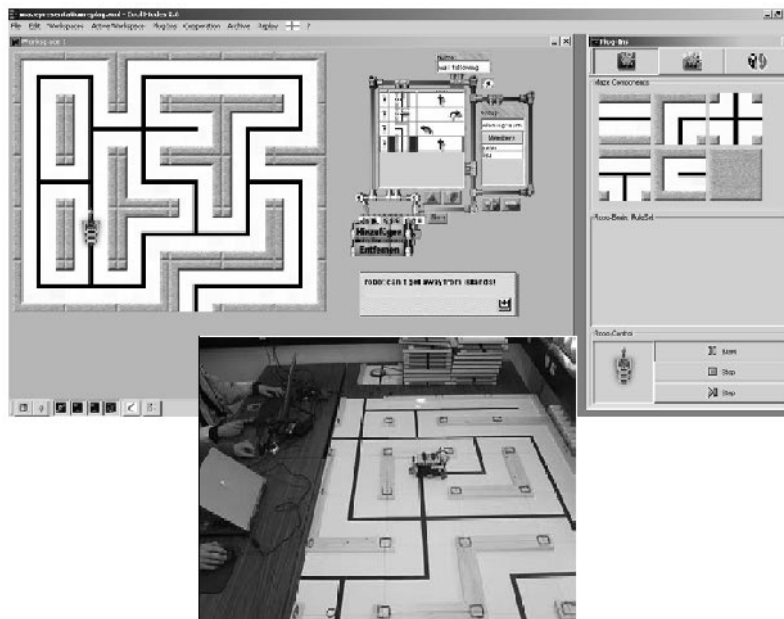


Abbildung 3: Roboter im Labyrinth - Cool Modes im COLDEX-Projekt

einsetzbar ist und innovative Formen der Kooperation ermöglicht. Im Rahmen von systematischen Evaluationen [Pin05] wurden Programmierer, die Modellierungssprachen für die Nutzung in Cool Modes entwickelt haben, sowie Lehrer, die das System im regulären Unterricht eingesetzt haben, zu ihren Erfahrungen befragt.

Die Interviews mit den Programmierern belegen, dass die Softwarearchitektur (welche die Reference Frame-Theorie umsetzt) eine deutliche Arbeitserleichterung und Zeitersparnis für die Programmierer mit sich bringt, z.B. hinsichtlich Serialisierung, Synchronisation, Design von Benutzerschnittstellen und Modellinteroperabilität. Herausragend ist, dass in keiner der Äußerungen Limitationen ausgedrückt wurden, die auf die Reference Frame-Theorie zurückführbar sind. Die Interviews mit den Lehrern zeigen die Eignung des Cool Modes Systems für pädagogische Anwendungen auf: in den Interviews wurden die mit Cool Modes interaktiv erstellbaren dynamischen Modelle ebenso positiv bewertet wie die Nutzung dieser Modelle als Basis für verschiedenste Gruppenlernformen im Unterricht.

Aktuelle Arbeiten beschäftigen sich damit, Cool Modes durch Anbindung an Modellarchive oder Integration mit anderen Tools über Lerndesigns in der Unterstützung asynchroner Kooperationsszenarien zu verbessern - die im System integrierten Funktionen zielen bisher vorwiegend auf synchrone Zusammenarbeit ab. Weitere laufende Forschungsaktivitäten haben es zum Ziel, aufgabenbezogene Analyse und intelligentes Feedback zu Modellen in Cool Modes zu ermöglichen - eine besonders in Lehr/Lernkontexten interessante Funktion, deren Umsetzung jedoch bei fehlender Eindeutigkeit von Lösungen eine Herausforderung darstellt.

## Literatur

- [Gaß03] Katrin Gaßner. Diskussionen als Szenario zur Ko-Konstruktion von Wissen mit visuellen Sprachen. Onlineveröffentlichung <http://www.ub.uni-duisburg.de/ETD-db>, 2003. Dissertation, Universität Duisburg-Essen.
- [Hop04] H. Ulrich Hoppe. Collaborative Mind Tools. In M. Tokoro und L. Steels, Hrsg., *A Learning Zone of One's Own - Sharing Representations and Flow in Collaborative Learning Environments*, Seiten 223–234, Amsterdam, Niederlande, 2004. IOS Press.
- [Jan03] Marc Jansen. MatchMaker TNG - a Framework to Support Collaborative Java Applications. In H. Ulrich Hoppe, Felisa Verdejo und Judy Kay, Hrsg., *Shaping the Future of Learning through Intelligent Technologies: Proceedings of the 11th International Conference on Artificial Intelligence in Education (AI-ED)*, Seiten 529–530, Amsterdam, Niederlande, 2003. IOS Press.
- [Jon00] David H. Jonassen. *Computers as Mindtools for Schools*. Prentice Hall, Upper Saddle River, NJ, 2000.
- [LMB<sup>+</sup>01] Akos Ledeczi, Miklos Maroti, Arpad Bakay, Gabor Karsai, Jason Garrett, Charles Thomason, Greg Nordstrom, Jonathan Sprinkle und Peter Volgyesi. The Generic Modeling Environment. In *Proceedings of the IEEE International Workshop on Intelligent Signal Processing (WISP)*, 2001. Online verfügbar unter <http://www.isis.vanderbilt.edu/Projects/gme/GME2000Overview.pdf>.
- [MMW98] K. Marriott, B. Meyer und K. B. Wittenburg. A Survey of Visual Language Specification and Recognition. In K. Marriott und B. Meyer, Hrsg., *Visual Language Theory*, Seiten 5–85, Berlin, 1998. Springer.
- [Pin05] Niels Pinkwart. *Collaborative Modeling in Graph Based Environments*. dissertation.de - Verlag im Internet, Berlin, 2005.
- [SH03] Daniel D. Suthers und Christopher D. Hundhausen. An Experimental Study of the Effects of Representational Guidance on Collaborative Learning Processes. *Journal of the Learning Sciences*, 12(2):183–219, 2003.
- [vJdJL<sup>+</sup>05] Wouter van Joolingen, Ton de Jong, A. W. Lazonder, E. Savelsbergh und S. Manlove. Co-Lab: Research and development of an on-line learning environment for collaborative scientific discovery learning. *Computers in Human Behavior*, 21:671–688, 2005.

Dr. **Niels Pinkwart** wurde am 28. Februar 1974 in Orsoy (jetzt Rheinberg) geboren. Nach dem Besuch des Krupp-Gymnasiums in Duisburg studierte er als Stipendiat der Studienstiftung des Deutschen Volkes von 1995 bis 1999 in Duisburg Mathematik und Informatik und schloss dieses Studium mit dem 1. Staatsexamen ab. Von 2000 bis 2005 war er als wissenschaftlicher Mitarbeiter in der Arbeitsgruppe von Herrn Prof. Hoppe im Institut für Informatik und Interaktive Systeme an der Universität Duisburg-Essen angestellt. 2005 promovierte er dort zum Thema kollaborativ nutzbarer interoperabler visueller Modellierungssprachen. Seit dem Sommer 2005 ist Herr Dr. Pinkwart als Post Doctoral Fellow im Institut für Human Computer Interaction an der Carnegie Mellon University angestellt, dort arbeitet er an einem intelligenten Tutorensystem für juristische Argumentationsstrategien. Die Forschungsinteressen von Herrn Dr. Pinkwart liegen im Bereich der interaktiven und kollaborativen Systeme mit Anwendungen in Lehr/Lernszenarien. In diesen Gebieten ist er durch mehr als 30 Veröffentlichungen ausgewiesen und durch zahlreiche Projekte und Gutachtertätigkeiten in die internationale Forschung eingebunden.