

# Konzept eines Musiktheorie-Servers

Jörg Garbers

Interdisziplinäre Forschungsgruppe KIT-MaMuTh

für Mathematische Musictheory

Technische Universität Berlin

<http://flp.cs.tu-berlin.de/MaMuTh>

## Abstract

This article shows, how well known and widely applied software techniques can support a scientific community of musicologists performing computer experiments. The scientific actions of performing an experiment, creating a protocol of the relevant events, publication, discussion, citation and reuse of results are discussed in the realm of computer experiments and linked to commonly known computer usage operations. Special emphasis is put on the assumption, that the success of these techniques highly depends on the motivation of a community to see common knowledge as a dynamically evolving set of interrelated propositions, which are not necessarily consistent and need a very careful way of citation.

In diesem Artikel geht es darum, wie die wissenschaftlichen Kulturtechniken des Experimentierens, des Protokollierens, des Verbreitens wissenschaftlicher Resultate, ihrer Weiterverwendung, Verbesserung, Zitierung und Auswertung, durch die Bereitstellung eines gemeinsamen Computers (Datei-/Webserver) und durch eine zu entwickelnde Nutzungskultur unterstützt werden kann.

Diese neu zu entwickelnde Kultur wird nötig vor dem Hintergrund einer wachsenden Zahl von Experimenten, die mit Hilfe des Computers durchgeführt werden. Zwar sollte das zentrale wissenschaftliche Ergebnis eines *Computereperiments* weiterhin in einem Artikel zusammengefaßt werden können, um es anderen Wissenschaftlern zu vermitteln, andererseits sind die Zwischenergebnisse (*Daten*) oft nicht sinnvoll durch Druckmedien wiederzugeben und nur mit Hilfe des Computers *weiterverwendbar* und verstehbar.

Nach einer Motivation für die Durchführung von Computer-Experimenten (Abschnitt 1), wird vorgestellt, wie ein Experiment protokolliert (Abschnitt 2) und durch andere Forscher wiederholt, dupliziert und verändert werden kann (Abschnitt 3). Das Identitäts- und Zitierproblem von sich zeitlich mit wachsendem Wissen ändernden Dateien (Abschnitt 4 und 5) wird im Rahmen der Versionsverwaltung und des Versionenvergleichs in der Softwareentwicklung und -nutzung besprochen (Abschnitt 6) und auf unterschiedliche Datenformate bezogen (Abschnitt 7). Zum Schluß geht es um praktische Fragen der *Navigation* im gemeinsamen kulturellen Raum eines Dateiarchivs, das von einer Sammlung verschiedener Experimente und ihrer Resultate aufgespannt und von Institutionen und Nutzern evaluiert wird (Abschnitt 8).

## **1 Was sind Computer-Experimente?**

Computer sind Universalmaschinen, die, gefüttert mit unterschiedlicher Software, unterschiedliche Aufgaben erfüllen können. Als „Office“-Hilfe sind sie aus Büros kaum noch wegzudenken und dienen mit Internetanschluß als Informations- und Kommunikationsmittel.

In den *Geisteswissenschaften* beginnt sich der Computer als zweifaches Werkzeug zu etablieren. Auf der einen Seite besticht er dadurch, daß er in kürzester Zeit große Datenmengen durchsuchen und auswerten kann und dadurch dem Menschen stupide oder ansonsten nicht zu bewältigende Arbeit abnimmt. Auf der anderen Seite kann man sich seine

unbestechliche Exaktheit zunutze machen um *operationalisierbare* Theorien in zwei Schritten zu überprüfen. Der erste Schritt, eine Theorie auf dem Computer zu implementieren, ist bereits der Beweis dafür, daß die Theorie operationalisiert, d.h. exakt formuliert ist. Der zweite Schritt, die Theorie selbst auf ihre Korrektheit hin zu überprüfen, besteht darin, das Computerprogramm mit unterschiedlichen Daten zu füttern und zu prüfen, ob die Ergebnisdaten mit der Erwartung übereinstimmen.<sup>1</sup>

Dabei ist entgegen bestehenden Vorurteilen zu bemerken, daß die operationalisierbaren Theorien nicht unbedingt die weniger interessanten Theorien sind. In der *Computerlinguistik* etwa werden Theorien einerseits operationalisiert, um sie in Anwendungen der automatischen Sprachverarbeitung verfügbar zu machen, andererseits zeigt sich aber auch, daß durch den Zwang zum Operationalisieren einige Phänomene der Sprache erst richtig erkannt werden und sich so interessante neue Forschungszweige eröffnen.

Die Ergebnisse reiner Computorexperimente, nämlich *Daten, die nur auf anderen Daten und einer Berechnungsvorschrift basieren*, haben gegenüber Experimenten, in denen menschliche Kompetenz stark involviert ist, den Vorteil, daß sie reproduzierbar und damit prinzipiell intersubjektiv verifizierbar sind. Daß in den Ausgangsdaten von Menschen kodiertes Wissen stecken kann, ist dabei nicht ausgeschlossen. So wird eine Theorie der Umwandlung von phonetisch transkribierten Lautsignalen in Text sinnvollerweise zunächst mit von einem Menschen kodierten und geprüften Daten getestet. Später kann die Umwandlung von Lautsignalen in phonetischen Text von einem anderen Programm geleistet werden und beide Programme zusammen können so als Anwenderprogramm zum Diktieren angesehen werden. Diese Verknüpfung und Weiterverwendung von Theorien wird uns später (Abschnitt 5) noch beschäftigen.

## **2 Das Protokoll zu einem Computer-Experiment**

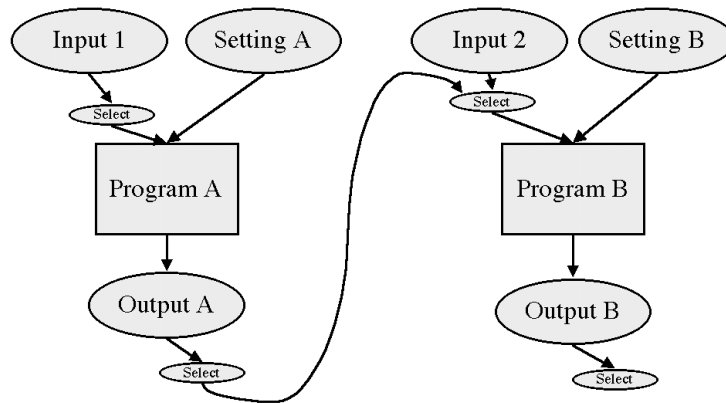
Damit ein Experiment technisch und inhaltlich nachvollzogen werden kann, gibt es in der experimentellen Physik zwei Arten von Protokollen: *das Verlaufs- und das Ergebnisprotokoll*. Das Verlaufsprotokoll enthält Informationen über die verwendeten Geräte, deren Genauigkeit, die Reihenfolge und den zeitlichen Ablauf der Beobachtungen, sowie die Meßdaten und die Namen der verantwortlichen Experimentatoren. Das Ergebnisprotokoll enthält eine Darstellung der zugrunde gelegten Theorie, Querverweise, eine Auswertung der Daten und eine Schlußfolgerung für die Theorie.

Im folgenden wollen wir als *Protokoll eines Computorexperiments* die Angabe der Eingangs- und Ergebnisdaten, der benutzten Programme, der Programmeinstellungen und ggf. die Folge der Benutzeraktionen ansehen, die zu den Ergebnisdaten führen. Dies entspricht in etwa dem Verlaufsprotokoll in der Physik. Die Erstellung eines solchen Protokolls kann vom Computer unterstützt werden.<sup>2</sup>

---

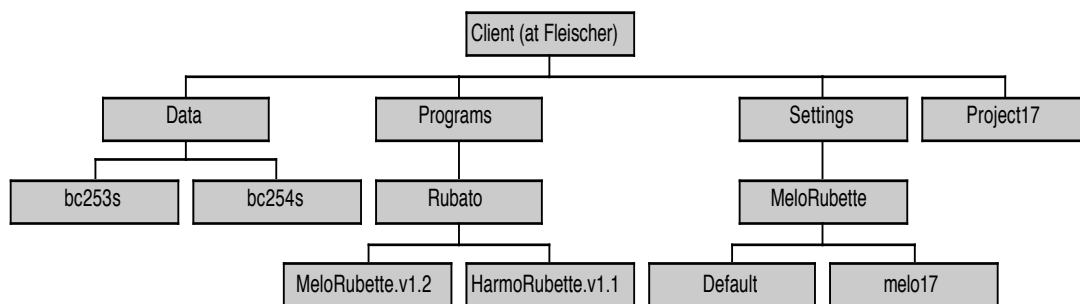
<sup>1</sup> Dies ist eine vereinfachte Situation. Tatsächlich hat man manchmal (etwa bei der HarmoRubette) nur eine grobe Vorstellung davon, was das Ergebnis einer neuen Theorie sein wird. In diesem Fall helfen Plausibilitätschecks mit einfach konstruierten Beispielen die Programmierfehler zu eliminieren. Danach kann man in komplizierteren Beispielen explorativ ein Gefühl für das differenzierte Verhalten der Theorie entwickeln.

<sup>2</sup> Auf dem Apple-Computer können die Nutzeraktionen z.B. durch Apple-Script protokolliert werden.



**Abbildung 1** Ein typisches Computerexperiment

Das untenstehende Diagramm (Abb. 2) geht davon aus, daß die Daten, Programme, und Programmeinstellung als *Dateien* vorliegen. Die Metainformation der am Experiment beteiligten Dateien und die Liste der ausgeführten Benutzeraktionen seien in der *Projektdatei* festgehalten. Die Existenz dieser Dateien wird in den nachfolgenden Abschnitten vorausgesetzt, denn nur so können diese an einen anderen Rechner übertragen und das Experiment reproduziert werden.



**Abbildung 2** Dateiverzeichnis eines Computerexperiments

### 3 Vermittlung, Weiterverwendung und Zitierung von Experimentierdateien

Das im vorigen Abschnitt vorgestellte Protokoll verfolgt den Zweck, anderen Forschern das Experiment zu *vermitteln* und es ihnen zu ermöglichen, mit Hilfe des Protokolls und der im Protokoll aufgeführten Dateien das Experiment auf ihrem eigenen Rechner nachzuvollziehen oder zu modifizieren und eigene Forschung zu treiben. Computerexperimente sind somit fast ohne Aufwand reproduzierbar und der Versuchsaufbau *weiterverwendbar*. Dies ist mit Experimenten in anderen Disziplinen, die physisch existierende Objekte (wie z.B. Menschen oder einen Protonenbeschleuniger) voraussetzen, bisher kaum möglich gewesen.<sup>3</sup>

Das Kopieren und Verändern von Dateien erzeugt jedoch ein mehrschichtiges Verwaltungsproblem, dem wir uns im folgenden zuwenden wollen und das eng mit der Fragestellung verbunden ist, was die *Identität einer Datei* ist und wie in Zukunft *Daten zitiert* werden. Mit der Identität eines Buches meinen wir im allgemeinen nicht die physische Identität, sondern das, was man beim Zitieren als Basis für eine Seitenzahl angibt, nämlich

<sup>3</sup> In der Physik wird gefordert, daß Experimente mit anderen Geräten reproduzierbar sind, wenn diese die gleichen Eigenschaften wie die im Protokoll beschriebenen Geräte haben. Darum sind physikalische Experimente im allgemeinen reproduzierbar, wenn auch manchmal nur mit großem Aufwand.

Autor, Titel und Auflage des Buches. Bereits das Weglassen der Auflagennummer kann (aber muß nicht) zu fehlerhaften Zitatstellen führen.

Wir wollen nun betrachten, inwiefern Texte im World-Wide-Web mit Büchern in zitatologischer Hinsicht vergleichbar sind. *WWW-Adressen* haben die Eigenschaft, auf Texte oder gar Textstellen im Internet zu verweisen und es gibt Hilfsmittel (Browser) mit deren Hilfe man sich die Texte beschaffen kann. Im Gegensatz zu gedruckten Büchern sind Texte im Internet oft nicht stabil: Sie werden verändert, aktualisiert, verschwinden oder ändern Ihre Adresse, der WWW-Server ist manchmal nicht erreichbar, etc. In diesem Sinne ist das World-Wide-Web wie ein großes *dynamisches Buch*, das ständig in neuen Auflagen erscheint, wobei die alten Auflagen vom Markt verschwinden und die Zitate von Abschnitten nur nach und nach aktualisiert werden.

Ob die Inhalte im WWW dynamisch oder statisch sind, hängt von den Informationsanbieter ab (dies sind Personen, Firmen, Institutionen usw.). So garantieren manche Anbieter, daß die Inhalte zitierbar bleiben, d.h. daß die Texte in einem Archiv *fixiert* aufbewahrt werden. Für verlässliches wissenschaftliches Arbeiten braucht die Gemeinde der Forschenden (insbesondere die der Computer-Experimentatoren) also ein institutionell betriebenes Archiv für weiterverwendbare und zitierbare Dateien.

#### **4 Organisation eines Datei-Archivs**

Archive haben die Eigenschaft, daß man sich in ihnen gut oder schlecht zurechtfinden kann. Zusammengehörige Daten<sup>4</sup> sollten als solche erkennbar sein. Hier sollen einige der mit dem Werkzeug CVS<sup>5</sup> möglichen Gruppierungen und über CVS hinausgehende Zusammenfassungen von Dateien aufgezeigt werden.

CVS ist ein Versionsverwaltungssystem für Dateien mit folgender Semantik: Eine Datei im Computer (z.B. ein Text) hat eine *zeitlich veränderliche Existenz*. Der Dateiname bleibt dabei trotz Veränderung des Dateiinhalts derselbe und zeigt die den einzelnen Versionen des Textes übergeordnete Identität an.<sup>6</sup> CVS erlaubt es nun seinem Benutzer, von Zeit zu Zeit den aktuellen Zustand der Datei im CVS-Archiv einzuchecken und somit *einzufrieren*. Dadurch kann der Benutzer später aus dem Archiv eine ältere Version der Datei rekonstruieren, etwa wenn er merkt, daß die seitdem vorgenommenen und möglicherweise eingeeckten Änderungen an der Datei schlecht waren.

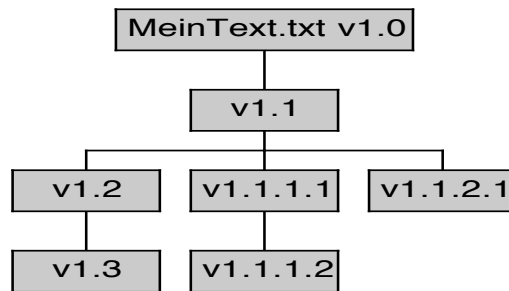
Innerhalb des Archivs werden nicht nur alle Versionen der Datei aufbewahrt, sondern es wird zudem festgehalten, welche Version der Datei durch Veränderung *aus welcher anderen Version hervorgegangen* ist. Z.B. wurde in folgendem Diagramm die Datei „MeinText.txt“ an CVS übergeben (Version 1.0), dreimal verändert eingeeckert (v1.1 bis v1.3), dann basierend auf v1.1 eine neue Variante (v1.1.1.1 bis v1.1.1.2) entwickelt, auch diese verworfen und erneut ausgehend von v1.1 die Version v1.1.2.1 entwickelt.

---

<sup>4</sup> Da es immer *mehrere Kategorien der Zusammengehörigkeit* gibt, gibt es die Archivwissenschaften, die sich um den Aufbau von Archiven kümmern.

<sup>5</sup> CVS: “Concurrent Versions System”. Siehe <http://www.sourceforge.com/CVS> und <http://www.loria.fr/~molli/cvs-index.html>.

<sup>6</sup> So wie auch die Identität eines Menschen unabhängig von der Mahlzeit, die er zu sich nimmt, verstanden wird.



**Abbildung 3** Versionsbaum einer Datei

Tatsächlich kann so ein *Entwicklungsbaum* auch anders entstanden sein, nämlich indem zunächst der Autor von v1.1 diese Version verändert als v1.2 eingchecked hat, dann ein anderer Autor v1.2 als weniger brauchbar als v1.1 empfand und seine eigenen Änderungen an v1.1 als v1.1.1.1 eingchecked hat. Danach haben die Autoren ihre zwei Alternativen in zeitlich irrelevanter Reihenfolge weiterentwickelt.

In praktischer Hinsicht wichtig bei solcherlei Verzweigungen ist, daß die Gemeinde der Nutzer ab und zu *Hauptzweige* auszeichnet, die als verlässliche Basis der Weiterentwicklungen dienen können (siehe Abschnitt 8). Häufig sind die Entwicklungen in parallelen Zweigen auch orthogonal zueinander, so daß sie mit einem Tool (CVS-Merge) vereinigt und die Vereinigung als neuer Hauptzweig definiert werden kann.

CVS bietet noch einige andere Funktionen an, die die *Koordination* von Editionen mehrerer Benutzer untereinander unterstützt<sup>7</sup>, etwa die Möglichkeit zu prüfen, ob eine Datei, die man sich aus dem Archiv geholt hat, mittlerweile in einer *aktualisierten* (möglicherweise verbesserten) Version im Archiv vorliegt. Auch kann man sich für Textdateien die *Differenz zweier Versionen* anschauen, d.h. die Aufmerksamkeit gleich auf die zuletzt editierten Zeilen lenken, ohne beide Texte vollständig lesen zu müssen.

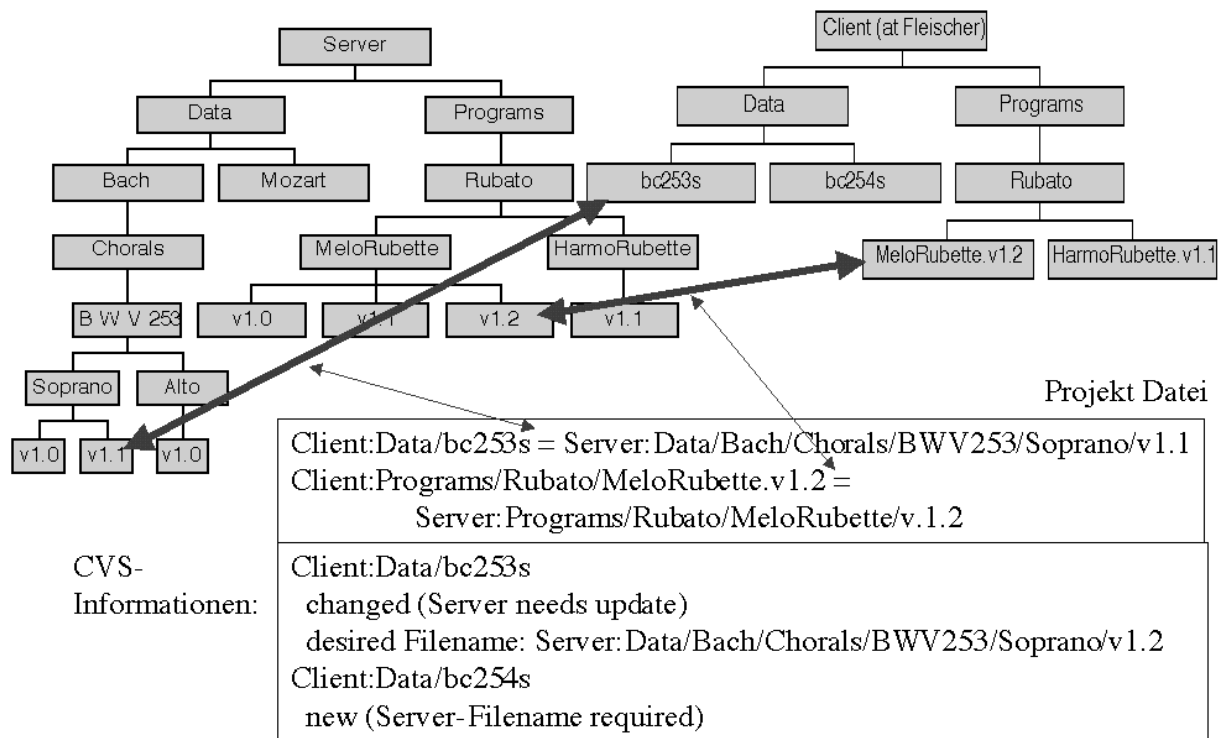
## **5 Wiederverwendung und Zitieren von Experimentierbestandteilen**

Das folgende Diagramm veranschaulicht, wie ein Experiment im gemeinsamen CVS-Archiv so lokalisiert werden kann, daß die Wiederverwendung von Dateien und Unterschiede zu anderen Experimenten deutlich werden. Da zur *Überprüfung eines Computerexperiments die Überprüfung aller seiner Bestandteile* gehört, ist es wichtig von der Wiederverwendung von Komponenten Gebrauch zu machen, da man deren *Korrektheit dann nur einmal überprüfen* muß.

Ein Ordner auf dem Rechner des Forschers (Client) beinhalte alle für das Experiment benötigten Dateien und die Projektdatei, die Aufschluß darüber gibt, welche Dateien aus dem Archiv kopiert, welche neu angelegt wurden und welche Benutzeraktionen zum Anlegen notwendig waren (Protokoll). Der lokale Ordner (rechts) bildet somit eine *lokale Karte*<sup>8</sup> (obere Hälfte der Projektdatei) mit allen projektrelevanten Dateien des globalen Archivs (links).

<sup>7</sup> Gegenüber anderen sogenannten Groupware- und Workflow-Programmen hat CVS den Vorteil frei verfügbar zu sein. Es wäre aber trotzdem sinnvoll zu prüfen, ob einige der in diesem Artikel skizzierten, über CVS hinausgehenden Nutzungsszenarios nicht bereits von einem anderen Programm unterstützt werden.

<sup>8</sup> Siehe auch Abschnitt 1.2.



**Abbildung 4** Das Einbetten lokaler Dateien ins Archiv

Wird nun ein neues Experiment veröffentlicht, heißt dies, alle geänderten oder neu erzeugten Dateien einschließlich der Projektdatei ans Archiv zu übergeben. Dabei können Informationen in der Projektdatei, die nur lokal von Interesse sind (z.B. die Dateinamen auf dem Client), entfernt bzw. durch die Archivbezeichnungen ersetzt werden.<sup>9</sup> Nicht veränderte Dateien behalten im Archiv ihre alte Identität (d.h. Versionsnummer).

Es ist zu erwarten, daß die Nichtbeachtung dieses Verfahrens, z.B. wenn jemand Dateien aus dem Archiv kopiert und diese beim einchecken des Experiments als neue Datei angibt, ähnlich *sanktioniert* werden wird, wie unterlassene Quellenangaben und *falsches Zitieren* in traditionellen wissenschaftlichen Texten. Ohne Disziplin bei der Nutzung des Archivs ist es schnell nicht mehr überschau- und navigierbar. Gerade wenn viele Daten produziert werden, ist eine halbautomatische *Falsifizierungshilfe*, die auf *protokollierten Abhängigkeiten* (Zitaten) basiert, wünschenswert. Wenn Datenquellen korrigiert werden, bedeutet dies ja, daß möglicherweise die auf den unkorrigierten Daten aufbauenden Schlußfolgerungen zu verwerfen sind. Wenn dagegen die ursprünglichen Quellen nicht mehr nachzuvollziehen sind, kann die im Archiv gespeicherte Projektdatei auch nicht als Falsifizierungshilfe herangezogen werden.

In diesem Zusammenhang müßte noch die *Verantwortbarkeit* von Ergebnissen diskutiert werden. Einerseits muß sich der Forscher natürlich auch weiterhin kritisch mit den Ergebnissen anderer Forscher auseinandersetzen, bevor er sie für sein Experiment als gegeben voraussetzt (indem er die fremden Dateien weiterverwendet), andererseits will er aber auch manchmal nur die *Differenz* seines Versuchsaufbaus zu einem vorherigen verantworten ohne jenen vollständig verstanden zu haben. Dieses legitime Verhalten ist auch Konsequenz der zunehmenden Arbeitsteilung in immer komplexer (und z.T. interessanter) werdenden

<sup>9</sup> Z.B. die Anmerkung im Protokoll "Habe bc253s mit Texteditor geändert" könnte in die Anmerkung "Habe <http://www.MaMuTh.de/CVS/Data/Bach/Chorals/BWV253/Soprano/v1.1> mit Texteditor geändert und v1.2 genannt" übersetzt werden.

Experimenten. Eine kritische Diskussion über die Verlässlichkeit einzelner Experimente und Daten in komplexen Projekten kann hier natürlich nicht geführt werden, sondern muß innerhalb der Forschergemeinde stattfinden (siehe auch Abschnitt 8).

## **6 Granularität der Wiederverwendung, Stabilität und Abstraktion in der Softwareentwicklung und Softwarenutzung**

Arbeitsteilung und Abstraktion von den Interna, wie ein Ergebnis bei anderen Entwicklern zustandekommt, sind die Grundpfeiler der Erfolge und Produktivitätszuwächse in der Informationstechnik. Gegenüber den Benutzern etwa, muß ein Programmierer nur das Verhalten, aber nicht das Innenleben eines Programmes rechtfertigen. Gegenüber Kollegen, die im Projekt mitarbeiten, muß er über Datenstrukturen und Programmierschnittstellen Rechenschaft ablegen. Die Situation von Programmierern und Anwendern kann auf andere Datenproduzenten und -konsumenten z.B. in der Forschung übertragen werden und soll deshalb hier mit Bezug auf CVS skizziert werden.

Die Benutzer eines Programmes wollen eine möglichst fehlerfreie, getestete Version haben. Für sie ist das Programm nur Mittel zum Zweck, bzw. muß im Verbund mit anderen Programmen *robust* sein. Sie werden sich nur für sogenannte *Releases*<sup>10</sup> der Binärversion und deren *Patches*<sup>11</sup> interessieren, die etwa in Monats oder Jahresabständen herauskommen. Sie benutzen und zitieren das Programm nur als *ganzes* und referenzieren keine Quelldateien, aus denen das Programm zusammengesetzt wurde.

Anders die Entwickler: Für sie besteht das Programm aus einer Hierarchie von Zugriffsebenen. Der Quelltext kann in mehreren Unterverzeichnisstrukturen organisiert sein, Unterprogramme in mehrfach verwendbare Module, diese in Dateien, diese in Schnittstelle und Implementation, lokale und globale Programmteile, etc. Um Codeteile zum Bau unterschiedlicher Programme zu verwenden ist es sinnvoll, alle diese Dateien einzeln in einem CVS-Archiv zu verwalten und in den *unterschiedlichen Kontexten* verschiedener Programme oder Programmversionen zu nutzen.<sup>12</sup>

Änderungen am Programm werden teils lokal in einer Quelldatei vorgenommen, z.T. kann aber auch eine logische Änderung mehrere Stellen betreffen, wenn etwa die Programmierschnittstelle für ein Modul geändert werden muß. Im allgemeinen ist ein Programm besser *wartbar*, je weniger Dateien für eine *logische Änderung* modifiziert werden müssen.<sup>13</sup> Allerdings versionieren Entwickler auch nicht immer ganz sauber: Typischerweise werden zwischen zwei Testläufen gleich mehrere logische Änderungen vorgenommen, oder es wird erst nach einer gewissen Zeit (z.B. jede Woche) die Gesamtheit der in der Zeit vorgenommenen Änderungen eingecheckt. Je seltener CVS benutzt wird, desto schlechter kann man später im allgemeinen anhand des Archivs feststellen, *weshalb* eine bestimmte Änderung vollzogen wurde, es sei denn, man macht von den *Kommentaren* gebrauch, die man an CVS mit einer geänderten Datei übergeben kann.

---

<sup>10</sup> Für den Benutzer ausgezeichnete Versionen.

<sup>11</sup> Korrekturen am Bestehenden, die keine neuen Merkmale und damit Fehlerquellen beinhalten sollen, sondern vorhandenes nur reparieren.

<sup>12</sup> Unter <http://www.FreeBSD.org/cgi/cvsweb.cgi> findet man ein Web-Interface zu dem von den Entwicklern des Betriebssystems FreeBSD benutzten CVS-Archiv.

<sup>13</sup> Darum ist es wichtig, in einer frühen Phase der Softwareentwicklung bereits eine saubere und leicht erweiterbare Softwarearchitektur zu haben.

## 7 Versionenvergleich auf Dateinhalt und Objekten

CVS unterstützt die Versionskontrolle auf der Ebene von Dateien bzw. Verzeichnissen von Dateien. Dort ist CVS sehr stark und leicht und schnell zu handhaben. Da Dateien eine typische Verwaltungseinheit auf dem Computer darstellen, ist dies sehr sinnvoll. Allerdings sind manchmal auch größere oder kleinere Einheiten wünschenswert. Wenn eine Datei beispielsweise in *Datenblöcke* organisiert ist, interessiert man sich nur für die geänderten Datenblöcke und hätte die Änderungsinformation gerne dementsprechend präsentiert. CVS selbst bietet nur die Differenzbildung von Versionen einer *Textdatei* an.<sup>14 15</sup>

Als Programmierer interessiert man sich oft eher für andere Datenblöcke, nämlich solche, die den Einheiten der Programmiersprache entsprechen. Dann interessiert man sich nicht für veränderten bedeutungsleeren *Whitespace*<sup>16</sup>. Den zwei Versionen will man dann etwa unter dem Gesichtspunkt der Quelldatei in einem Softwareprojekt dieselbe *Identität* zuweisen.

Als *DenoteX*-Benutzer<sup>17</sup> interessiert man sich manchmal sowohl für eine didaktisch mit Layoutinformation organisierte *DenoteX*-Datei manchmal nur für dessen Normalform, in der Whitespace entfernt ist, jede Denotatordefinition auf einer Zeile steht und die Zeilen nach Denotatornamen sortiert sind. Dann können *DenoteX-inhaltliche Änderungen* von *Darstellungsänderungen* unterschieden werden. Wie man sieht, können inhaltlichen Vergleiche nur von an das Datenformat *angepaßten* Programmen durchgeführt werden.

Mit der weiteren Verbreitung der zum Nachfolger von HTML erkorenen Datenbeschreibungssprache *XML* ist zu erwarten, daß auch für den innerdateilichen Bereich von *XML*-Dokumenten Versionssysteme entstehen. Das Zitieren einzelner Strukturen innerhalb von *XML*-Dokumenten ist mithilfe der *XML-Pointer* Definition<sup>18</sup> bereits möglich. Eine sinnvolle Anwendung des *XML*-Standards ist es, die zwei Versionen einer nicht notwendigerweise im *XML*-Format vorliegenden Datei unter jeweils unterschiedlichen Aspekten (Layout oder Struktur oder Semantik) nach *XML* zu projizieren und dort *strukturell* zu vergleichen. Ein ähnlicher Ansatz wird in *Rubato* mit dem von Programmierklassen zu implementierenden *Denotatoren-Protokoll* verfolgt, das es einem Programmobjekt erlaubt, sich als Denotator in unterschiedlichen *Formräumen* zu zeigen und so strukturell mit den Projektionen von Objekten anderer Klassen verglichen werden kann.<sup>19</sup>

## 8 Evaluation, Reviews und Navigationshilfen für das Archiv

CVS bietet mit seinen Sicherheitsfunktionen und einem gewissen organisatorischen Zwang eine gute Grundlage für ein gemeinsam genutztes Dateiarchiv. Zusätzliche Werkzeuge zum Versionenvergleich können diese noch attraktiver machen. Entscheidend für den Erfolg als Unterstützungsinstrument für die wissenschaftliche Forschung ist jedoch, wie das Archiv von seinen Benutzern angenommen wird und ob sich eine *Zitier- und Kommunikationskultur* über und mit Hilfe des Archivs entwickelt, da es ein großer Unterschied ist, ob wissenschaftliche Ergebnisse und Experimente irgendwo gespeichert und prinzipiell zugänglich sind, oder ob

---

<sup>14</sup> In Textdateien sind die Datenblöcke durch das Zeilenendezeichen getrennt. Die Datenblöcke selbst enthalten nur Zeichen und keine Formatierungsinformation.

<sup>15</sup> CVS-intern wird aus Gründen der Speichereffizienz nur die Differenz zur Vorgängerversion gespeichert.

<sup>16</sup> Dies sind aus Layoutgründen hinzugefügte Leerzeichen, die keine semantische Information tragen und für die Erstellung des lauffähigen Programms keine Rolle spielen.

<sup>17</sup> Siehe Abschnitt 3.1 und <http://flp.cs.tu-berlin.de/MaMuTh/ifm/denotex>

<sup>18</sup> <http://www.w3.org/XML/Linking.html>

<sup>19</sup> Z.B. kann ein errechnetes Objekt als Denotator mit einem vom Benutzer editierten Denotator verglichen werden, der das vom Benutzer erwartete Ergebnis repräsentiert.



sie auch von anderen Forschern wahrgenommen werden. So wirkt sich in der *traditionellen wissenschaftlichen Kultur* der Bekanntheitsgrad des Autors und die Anzahl der Reviews stark darauf aus, ob seine Veröffentlichungen gelesen werden.

Ich möchte hier eine Mischung aus *Evaluationen* von Ergebnissen durch *Benutzer* und solchen durch *Institutionen* propagieren. Die durch die Benutzung von Dateien mühsam erworbenen *Benutzererfahrungen* sollten anderen Nutzern als Kommentar zu den Dateien zur Verfügung gestellt werden können. Wie dies aussehen könnte, präsentiert einer der führenden Online-Buchläden, Amazon. Neben echten *Reviews* im herkömmlichen Sinne können Leser (Käufer) des Buches für andere Interessenten Kommentare hinterlassen, es z.B. als unverständlich oder gelungen werten. Außerdem kann man eine *quantifizierte Bewertung* abgeben.

The image shows a screenshot of the Amazon.com product page for the book "Godel Escher Bach: An Eternal Golden Braid" by Douglas Hofstadter. The page includes a navigation bar with categories like BOOKS, MUSIC, VIDEO, and GIFTS. Below the navigation bar, there are links for "BOOK SEARCH", "BROWSE SUBJECTS", "BESTSELLERS", "FEATURED IN THE MEDIA", and "AWARD WINNERS". The main content area displays the book's title, author, and a "Book Information" section with links for "at a glance", "reviews", and "customer comments". The price is listed as \$16.00, with a 20% discount from the list price of \$20.00. The book's availability is noted as "Usually ships within 24 hours." The page also shows the book's cover, a "Keyword Search" box, and a "Write an online review" button. Hand-drawn arrows point from text labels on the left to specific elements on the page: "Institutionelle Reviews" points to "Book Information", "Nutzerkommentare" points to "customer comments", "Popularität" points to "Amazon.com Sales Rank: 216", "Wertung" points to "Avg. Customer Review: ★★★★★", and "Weitere Informationen" points to "Write an online review".

Abbildung 5 Leserevaluation von Büchern bei amazon.com

Die institutionelle Evaluation dient dazu, im Dschungel des Archivs verlässliche Daten auszuzeichnen und als gemeinsame Quelle für weitere Experimente vorzuschlagen, mit der Hoffnung, daß die mehrfache Benutzung institutionell propagierter Standarddateien deren Güte und die *Vergleichbarkeit* von Experimenten steigert. Dafür müssen Kriterien der Güte einer Datei, wie analytische Korrektheit, repräsentationelle Sauberkeit und Handhabbarkeit entwickelt werden. Die institutionelle Diskriminierung gleichberechtigter theoretischer Ansätze kann dabei durch die Nutzerevaluation (s.o.) ausgeglichen werden.

Auf einem Forschungs-Server könnten neben den Evaluationsdaten noch *Kategorien* wie z.B. die Nutzergruppe (Entwickler/Anwender) als Filter für die Navigation im Archiv vergeben werden, oder es könnte zu einer Datei automatisch aufgezeigt werden, in welchen Projektdateien sie *zitiert* wird. Weitere Ideen und Hilfsmittel zur Navigation im gemeinsamen Wissensraum sollten nach und nach entwickelt werden.

## 9 Zusammenfassung

Es wurde dargestellt, wie Computer-Experimente protokolliert und als Sammlung von Dateien anderen Forschern zur Überprüfung und Weiterverwendung übergeben werden kann. Die Foschergemeinde sollte dafür ein Dateiarchiv mit Versionskontrolle einrichten und weitere Hilfsmittel für den Datenvergleich auf innerdateilicher Ebene anbieten. Im Falle einer

regen Nutzung des Archivs sollte über Evaluationen und weitere Navigierhilfen nachgedacht werden.

Es muß festgehalten werden, daß die technischen Hilfsmittel zum Realisieren der in Abschnitt 2 geforderten *Kulturtechniken des wissenschaftlichen Experimentierens mit Computern* vorhanden sind. So existiert Musikexperimentiersoftware (z.B. *Rubato* und *humdrum*), Betriebssysteme und Programme bieten die Möglichkeit Benutzeraktionen zu protokollieren und zu wiederholen (z.B. *Apple-Script*), das Versionsverwaltungssystem *CVS* unterstützt die Verknüpfung von eigenen Dateien mit Archivdateien und mit Sourceforge<sup>20</sup> existiert sogar bereits ein integriertes Web-basiertes System für die Verwaltung von gemeinsamen Projekten. Damit lassen sich vermutlich viele der in der Softwareentwicklung verbreiteten Kulturen der Wiederverwendung gewinnbringend auf die geisteswissenschaftliche computergestützte experimentelle Forschung übertragen.

---

<sup>20</sup> Siehe <http://sourceforge.net> : "SourceForge is a free service to Source developers offering easy access the best in CVS, mailing lists, bug tracking, message boards/forums, task management, site hosting, permanent file archival, full backups, and total web-based administration."