

# Die Informatik oder eine Neuerfindung der Sprache

**François Bry**

Institut für Informatik der Ludwig-Maximilians-Universität München  
<http://www.pms.ifi.lmu.de/>

Es ist allgemein bekannt, dass die Informatik sich verschiedener „Sprachen“, unter anderem Programmiersprachen, bedient. Eine verbreitete Meinung ist, dass die Informatiksprachen wenig mit „menschlichen Sprachen“ gemeinsam haben. Informatiksprachen sind ja „Befehlssprachen“ zur Steuerung von Computern oder zur Spezifikation von Berechnungen. Im Gegensatz dazu sind „menschliche Sprachen“, in der Informatik auch „natürliche Sprachen“ genannt, Vehikel von Kulturen und werden verwendet, um unter anderem Gedanken und Gefühle auszudrücken. Künstliche Sprachen zur Gerätesteuerung oder zu Berechnungen einerseits, natürliche Sprachen zum Ausdruck von Gedanken und Gefühlen andererseits: Der Begriff „Sprache“ in seinen verschiedenen gegenwärtigen Auslegungen führt also unmittelbar zur Gegenüberstellung von Technik und Kultur, welche unsere Zeit prägt. Wird aber der Begriff „Sprache“ in der Informatik einerseits und in der Kultur andererseits wirklich unterschiedlich ausgelegt?

Dieser Artikel stellt einen Versuch dar, den eigentümlichen Umgang mit Sprachen in der Informatik zu schildern. Nicht nur Programmier- und Befehlssprachen werden dabei betrachtet, sondern auch allerlei oft als „Formalismen“, „Schemata“ oder „Modelle“ bezeichneten Sprachen, die von Informatikern entwickelt, verändert und nicht zuletzt verwendet werden. Einerseits sind solche „technischen Sprachen“ viel einfacher als die „menschlichen Sprachen“. Andererseits ergänzen sie für viele ihrer Benutzer in schon sehr bedeutsamer Weise die „menschlichen Sprachen“. Dieser Artikel stellt die These auf, dass die technischen Sprachen der Informatik grundlegend auf der menschlichen Sprachfähigkeit beruhen und eine Neuerfindung der „menschlichen Sprache“ darstellen, die das Versprechen einer wesentlichen Erweiterung der menschlichen Sprachfähigkeit in sich trägt.

Dieser Artikel ist wie folgt gegliedert. Ein erster Abschnitt bietet eine Annäherung an den Begriff „Sprache“ aus Sicht der Informatik: Was bedeutet „Sprache“ in der Informatik, wie stehen Informatiksprachen zu „menschlichen Sprachen“, wie zentral ist die Stellung von Sprachen in der Informatik? Ein zweiter Abschnitt ist der Nutzung von Informatiksprachen zur Wirklichkeitskonstruktion gewidmet, die zunehmend auch außerhalb von Technik, Verwaltung und Wirtschaft stattfindet. Ein dritter Abschnitt geht auf Ansätze der Informatik zu grafischen und Gebärdensprachen ein und stellt die Frage, welche Zukunft solche Sprachen in der Technik sowie im Allgemeinleben haben. Ein vierter Abschnitt ist dem ständigen und sehr schnellen Wandel der Informatiksprachen gewidmet, welcher dem Mensch eine ganz neue Dimension des Umgangs mit „menschlichen Sprachen“ eröffnen könnte. Ein fünfter Abschnitt untersucht Besonderheiten der Informatiksprachen, ihre Fähigkeit zum Rechnen und berechnet zu werden. Es wird gefragt, ob dies eine Besonderheit von Informatiksprachen oder vielmehr die Wahrnehmung einer verborgenen Fähigkeit der menschlichen Sprachfähigkeit ist. In einem abschließenden Abschnitt wird die These aufgestellt, dass die Informatik zum einen auf der menschlichen Sprachfähigkeit beruht, zum anderen das Konzept „Sprache“ neu erfindet und dadurch erweitert.

## 1. Eine Annäherung an den Begriff „Sprache“ aus Informatiksicht

In der Informatik werden künstliche Sprachen entwickelt und verwendet, um sowohl Daten als auch Programme auszudrücken. Die Unterscheidung zwischen „Daten“ und „Programmen“ ist traditionell und üblich<sup>1</sup>, wenn auch nicht uneingeschränkt zutreffend. Unter „Daten“ werden einfache und zusammengesetzte Werte verstanden, die verändert oder in Berechnungen verwendet werden können, zum Beispiel der Name eines Patienten, die Beschreibung seines Gesundheitszustandes, bisheriger Therapien usw. Unter „Programmen“ werden Beschreibungen von Verfahren zur Änderung oder sonstigen Verwendung von Daten verstanden. Programme realisieren Algorithmen, die als Berechnungen mathematisch formalisiert werden. Programme werden möglichst für unbestimmte Werte, also Datensätze, definiert, damit sie allgemein verwendbar sind. Zum Beispiel stellt  $(x+y)/2$  ein Programm zur Berechnung des arithmetischen Durchschnitts zweier Werte, die durch die Variablen  $x$  und  $y$  gegeben sind. Werden zum Beispiel Patientendaten verwaltet — etwa von Fehlern befreit, mit neuen Befunden ergänzt, als Grundlage für Mitteilungen oder Rechnungen verwendet —, so werden diese Operationen des täglichen Leben von Informatikern unter Verwendung von Algorithmen als Berechnungen formalisiert und mittels Programmen realisiert.

Die Unterscheidung zwischen „Daten“ und „Programmen“ ist aus zwei Gründen nur bedingt zutreffend. Zum einen können — und werden! — Daten manchmal als Programme angesehen werden. Ein Bankkonto, bestehend zum Beispiel aus dem Namen des Kontoinhabers, dem Kontostand, und einer Ein- und Auszahlungsliste, kann sehr wohl als eine Art Programm angesehen werden, welches durch Überweisungen seinen eigenen Stand sowie den Stand von anderen Konten verändern kann. Diese Sicht ist keineswegs willkürlich oder ein Kuriosum. Vielmehr führt sie zum Kern von Ansätzen der Programmierung sowie deren mathematischen Grundlagen, die sich als nützlich erwiesen haben. Die Unterscheidung zwischen „Daten“ und „Programmen“ ist zum anderen auch insofern unzutreffend, dass Programme selbst durch (weitere oder sogar in manchen Fällen dieselben) Programme verarbeitet werden. Dies ist beispielsweise der Fall, wenn ein Texteditor, also ein erstes Programm, verwendet wird, um ein zweites Programm zu verändern. Es geschieht aber insbesondere immer, wenn Programme ausgeführt werden. Dazu werden spezielle Ausführungsprogramme verwendet — unter anderem so genannte Übersetzer (oder Compiler)<sup>2</sup> zur Erzeugung eines ausführbaren Programms in der Maschinensprache des ausführenden Computers.

Wie zutreffend oder unzutreffend die Dualität Daten/Programm auch sein mag, zeigt sie eine erste Ähnlichkeit zu den „menschlichen Sprachen“. „Menschliche Sprachen“ sind ja immer verwendet worden, zum einen um etwas festzuhalten (Gedanken, Gefühle oder auch Verwaltungsinformation), zum anderen um etwas zu bewirken (Anweisungen übermitteln oder Gefühle entstehen lassen). In dieser Hinsicht werden also Informatiksprachen genauso wie „menschliche Sprachen“ verwendet, um etwas festzuhalten oder etwas oder jemanden zu beeinflussen.

Die Informatik entwickelt und verwendet Sprachen, um möglicherweise komplexe Zusammensetzungen von Daten, wie etwa Patienten- oder Kontodaten, zu beschreiben und auch, um Algorithmen zu spezifizieren. Die Sprachen der Informatik können folglich in einer ersten Annäherung als „mathematische Sprachen“ angesehen werden. In der Tat sind die ersten Sprachen der Informatik in den 1950-er Jahren als Variationen und Erweiterungen von mathematischen Notationen entstanden.<sup>3</sup> Obwohl die Informatik seit diesen Anfängen viele

---

<sup>1</sup> S. (Sethi 1996), (Scott 1997) und (Sebesta 2003).

<sup>2</sup> S. (Dewdney 1993) Kapitel 53.

<sup>3</sup> S. (Moreau, 1987) und (Kidwell and Ceruzzi, 1994).

andere Sprachen und „Sprachparadigmen“, wie die verschiedenen Arten von Informatiksprachen heute genannt werden, entwickelt hat, die wesentlich von der „Sprachkultur der Mathematik“ abweichen, bleiben Informatiksprachen weitgehend „mathematisch“. Dafür gibt es einen einfachen Grund: die Automatisierung, der die Informatiksprachen unterworfen sind und die sie ermöglichen sollen.

Die Feststellung, dass Informatiksprachen „mathematisch“ sind, könnte zu dem Schluss verleiten, dass Informatiksprachen wenig mit „menschlichen Sprachen“ gemeinsam haben. Aber im Gegenteil, ihre Verwurzelung in der Sprach- und Ausdruckskultur der Mathematik, die sie weiterführen und ergänzen, ist ein Beleg der Nähe der Informatiksprachen zu den „menschlichen Sprachen“. Die Mathematik, insbesondere die Logik, folglich auch ihre Sprach- und Ausdruckskultur, sind ja bekanntlich aus Versuchen entstanden, Unzulänglichkeiten der „menschlichen Sprachen“ zu überwinden. Auch die Informatiksprachen stehen in dieser Tradition der mathematischen sowie auch philosophischen Logik.

Informatiksprachen sind, wie „menschliche Sprachen“, linear in dem Sinne, dass sie als eindimensionaler Text geschrieben werden. Es ist davon auszugehen, dass sich die Schriften der „menschlichen Sprachen“ in Anlehnung an das Gesprochene linear entwickelt haben. Die meisten mathematischen Notationen sind ebenfalls linear, wenn auch mit Zusätzen wie Unter- und Oberschriften. Der Logiker Gottlob Frege<sup>4</sup> hat eine „Begriffsschrift“ genannte Sprache für die mathematische Logik entwickelt, die zweidimensional, also „diagrammatisch“ und beinahe „grafisch“ ist. Vermutlich weil sie nicht linear war und daher für Logiker schwierig zu schreiben und für Typographen schwierig zu setzen war, hat sich die „Begriffsschrift“ trotz eindeutiger Vorteile nicht etabliert.

Kein anderer Grund als die Ähnlichkeit mit „menschlichen Sprachen“ erklärt die Linearität von Mathematik- und Informatiksprachen. Diese Ähnlichkeit ist von Belang, weil Informatiksprachen bis in die jüngste Zeit wie „menschliche Sprachen“ linear geschrieben worden sind. Erst im letzten Jahrzehnt haben sich so genannte Programmierumgebungen verbreitet, womit Daten und vor allem Programme „grafisch“ unter Verwendung von beweglichen und erweiterbaren Diagrammen verfasst werden können.<sup>5</sup> Solche Diagramme erinnern an die „Begriffsschrift“ und beruhen zum Teil auf ähnlichen Prinzipien. Die Möglichkeiten, die der Computer bietet, eröffnen also neue Möglichkeiten, Sprachen zu schreiben, insbesondere mit „zweidimensionalen Schriftarten“. Werden in Zukunft solche Ansätze auf „menschliche Sprachen“ übertragen werden?

Zwei Besonderheiten der „Informatiksprachlichkeit“ sind erwähnenswert, die den Bezug der Informatik zu den „menschlichen Sprachen“ weiter beleuchten: eine bemerkenswerte „Polyglottie“ und eine Vorliebe für „Zirkularität“.

**Polyglottie.** Informatiker bedienen sich vieler, sehr vieler Sprachen. Typischerweise werden in Software-Projekten Sprachen zur Beschreibung der Software-Komponenten und ihrer Interaktionen eingesetzt, Sprachen zur Beschreibung der Daten und ihrer Zusammensetzung, Sprachen zur Beschreibung der eigentlichen Operationen, die die Software realisieren soll, Sprachen zur Dokumentation, das heißt Erklärung, der Software, usw. Sogar der Ablauf der Software-Entwicklung wird mit Hilfe von sprachähnlichen Formalismen dokumentiert. Informatiker, sowohl in der Berufspraxis wie in der Forschung, sind regelrechte Sprachnarren!

Es hat in der Informatik mehrmals Anstrengungen gegeben, die Informatikpolyglottie ihrer gewaltigen Kosten wegen einzudämmen. Einer der bekanntesten dieser Versuche war ein

---

<sup>4</sup> 1848-1925, s. (v. Kutschera, 1989).

<sup>5</sup> S. (Poswig, 1996).

Auftrag des US-Verteidigungsministeriums, des „Department of Defense (DoD)“, in den 70-er Jahren des 20. Jahrhunderts, aus dem die Programmiersprache Ada<sup>6</sup> entstand. So wie dieser sind auch alle anderen Versuche, ein „Esperanto der Programmierung“ zu schaffen, insofern gescheitert, dass sie lediglich weitere Sprachen und keine einsprachige Informatik hinterlassen haben. Interessanterweise liegt ein wichtiger Grund dafür darin, dass immer wieder neue Anwendungsfelder für Informatiksprachen entstanden sind und weiter entstehen, die Anlass zu ganz neuen Sprachparadigmen oder auch zu Variationen vorhandener Paradigmen geben. Mit neuen Sprachen oder Dialekten lassen sich eben neue Anwendungsfelder oft viel leichter besetzen als mit älteren Sprachen.

Die Sprachvernarrtheit der Informatiker und die „Babel-Erscheinung“ in der Informatik sind weitere Hinweise auf die Nähe der Informatiksprachen zu den „menschlichen Sprachen“. Das Interesse für Sprachen ist in jeder Kultur belegt und Babel ist Parabel für die Vielfalt der sich voneinander abgrenzenden Kulturen. Immer wieder sind mit neuen Kulturen neue Dialekte oder Sprachen entstanden. Die Vorliebe der Menschen aller Kulturen für Sprachen ist nicht zu leugnen. Diese Vorliebe ist so allgegenwärtig, dass sie oft übersehen wird. Nur durch diese Vorliebe lassen sich ziemlich irrationale Auseinandersetzungen — wie jüngst in deutschsprachigen Ländern über die Rechtschreibreformen — erklären. Ein wesentlicher Unterschied, vielleicht sogar der Hauptunterschied zwischen Babel in der Informatik und in den Kulturen, liegt in der Schnelligkeit, mit der neue Dialekte und Sprachen entstehen. Kulturen brauchen Jahrhunderte, um neue Sprachen hervorzubringen. In der Informatik reichen dafür ein paar Jahre aus. Dies liegt zweifelsohne an der eingeschränkten Natur der Informatiksprachen, aber nicht ausschließlich daran. Es liegt zum einen auch an der relativ kleineren „Schreiberzunft“, die bestimmte Informatiksprachen verwendet und Abwandlungen davon entwickelt. Zum anderen liegt es auch an den technischen Grundlagen der Informatiksprachen: Sie sind auch deswegen relativ leicht zu erlernen und zu verwenden, weil sie automatisiert sind. Der Computer ist ein Sprachlehrer, der unermüdlich Sprachfehler mitteilt und auf mögliche Korrekturen hinweist.

**Zirkularität.** Mit dem Begriff „Zirkularität“ ist hier gemeint, dass Informatiksprachen oft und vorzugsweise „in sich selbst“ definiert und zum Großteil programmiert werden. Die Zirkularität hat einen praktischen Vorteil sowohl für die menschlichen Benutzer der Sprache wie auch für die Automatisierung, oder „Implementierung“, der Sprache. Sie ermöglicht, dass die Benutzer einer Programmiersprache ohne Bezug auf eine weitere Sprache die Programmiersprache lernen und verwenden können. Dies ist selbstverständlich nur deswegen möglich, weil dabei auch eine „menschliche Sprache“ verwendet wird. Die Zirkularität einer Informatiksprache vereinfacht zudem ihre Implementierung sehr wesentlich, weil abgesehen von einem eng begrenzten Sprachkern die Sprache „in sich selbst“ programmiert werden kann. Dieser Sprachkern wird direkt in den einfachen Maschinensprachen der Computer-Prozessoren implementiert, so dass die Maschinensprachen eine ähnliche Rolle bei der Implementierung spielen wie die „menschliche Sprache“ beim Erlernen der Informatiksprache durch einen Menschen. Zirkularität muss als weiterer Beleg der Nähe von Informatik- und „menschlichen Sprachen“ angesehen werden: Die Zirkularität ist ja ein allgegenwärtiges Merkmal aller „menschlichen Sprachen“, die alle als „ausreichend“ für die menschliche Kommunikation empfunden werden — oder vielleicht eher ausreichend dafür entstanden sind. Die Zirkularität ist ein grundlegendes Mittel zur Vermittlung einer „menschlichen Sprache“, wie unter anderem (einsprachige) Wörterbücher und Grammatiken belegen.

Diese kurze Betrachtung des Begriffs „Sprache“ aus Informatiksicht soll mit der Feststellung abgeschlossen werden, dass Sprachen in der Informatik allgegenwärtig sind. Es gibt kein

---

<sup>6</sup> S. (Nagl, 1998).

Problem der praktischen oder theoretischen Informatik, welches nicht mit Sprachen angegangen wird. Es gibt keinen Ansatz in der Informatik, der nicht mit Sprachen formalisiert wird. Sprachen haben in der Informatik eine zentrale Stellung inne — so wie „menschliche Sprachen“ in den Kulturen.

## **2. Informatiksprachen zur Wirklichkeitskonstruktion**

Informatiksprachen werden zur Definition und Vermittlung von Abstraktionen verwendet, die an erster Stelle an den Menschen gerichtet sind, nicht an den Computer. Gewiss werden Programme automatisch ausgeführt. Programme dienen allerdings vorrangig dazu, Verfahren so zu spezifizieren, dass sie von Menschen verstanden und auf Richtigkeit überprüft werden können. Der Vorrang des menschlichen Lesers von Programmen gegenüber den ausführenden Computern ist unter anderem darin erkennbar, dass die historische Entwicklung von Informatiksprachen vom Streben nach so genannten „höheren“ Ansätzen geprägt ist. „Höhere“ Sprachen oder Sprachparadigmen bezeichnen in der Informatik Sprachansätze, die den Menschen leicht zugänglich sind. Die Informatik widmet sich der Konstruktion von automatisch ausführbaren „Symbolwelten“ primär für Menschen.

Die „Symbolwelten“, die durch die Spezifikation von Daten und Programmen konstruiert werden, geben immer Wirklichkeitsausschnitte wieder. Beispiele solcher Ausschnitte sind in unserer technisierten Welt geläufig: Eine Arztpraxis verwendet zum Beispiel Patientendaten, eine Behörde verwendet Daten über natürliche oder juristische Personen, usw. Die Wahl eines Wirklichkeitsausschnitts und seine „Kodierung“, das heißt Darstellung, in Informatiksprachen stellen meistens eine Rekonstruktion des Beobachteten dar, welche für sich und unabhängig von der ebenfalls angestrebten Automatisierung für den menschlichen Nutzer wertvoll ist. Aus Unternehmen wird zum Beispiel immer wieder berichtet, wie ein Informatik-Projekt nebenbei und oft unerwartet zur Wahrnehmung von Entscheidungsstrukturen und/oder -prozessen im Unternehmen beigetragen hat. Umgekehrt, was der vorherigen Bemerkung nicht widerspricht, führt die Automatisierung durch Software oft zur Verschleierung von Entscheidungsprozessen in Unternehmen. Als zur Jahrtausendwende aus Furcht vor Datumsfehlern<sup>7</sup> sämtliche Software in Unternehmen und Behörden überprüft und erneuert wurde, wurden viele in Vergessenheit geratenen Entscheidungsprozesse in Erinnerung gebracht. Darauf folgend sind einige dieser Entscheidungsprozesse in Frage gestellt oder in einigen Fällen verändert worden.

Die Nutzung der Informatiksprachen zur Wirklichkeitskonstruktion ist ein weiterer Beleg der Ähnlichkeit von Informatiksprachen mit den „menschlichen Sprachen“.

Nicht unwesentlich ist, dass Informatiksprachen sich viel leichter als „menschliche Sprachen“ umformen oder neu definieren lassen, um neue Möglichkeiten der Wirklichkeitskonstruktion zu eröffnen. Zwei Beispiele dafür seien hier erwähnt: die Programmiersprachen Logo<sup>8</sup> und StarLogo.<sup>9</sup> Das Programmierparadigma von Logo ist die Steuerung eines Schreibgerätes, bildhaft „Schildkröte“ genannt. Die Programmiersprache bietet einfache und prägnante Sprachkonstrukte, deren Verwendung in natürlicher und didaktischer Weise in grundlegende Begriffe der Programmierung wie Iteration und Rekursion einführt. Eine Iteration wird zum Beispiel im Grunde wie folgt ausgedrückt: „4-mal wiederholen: 3cm geradeaus, dann 90°

---

<sup>7</sup> Die verbreitete Kodierung von Jahreszahlen durch die letzten zwei Ziffern lässt zum Beispiel nicht zwischen 1908 und 2008 unterscheiden.

<sup>8</sup> S. (*Senftleben, 1984*).

<sup>9</sup> S. (*Resnick, 1997*).

linksum“. StarLogo beruht auf dem Programmierparadigma von Gemeinschaften unabhängig agierender „Agenten“, bildhaft „Ameisen“ genannt, wobei die Selbständigkeit jedes Agenten durch Zufall simuliert wird. So kann in StarLogo ein Kreis mit einem Radius von 1cm dadurch definiert werden, dass eine große Anzahl von „Ameisen“ zuerst zu einem Punkt, dem Zentrum des Kreises, gebracht und dann um 1cm davon entfernt werden. Da jede „Ameise“ die Richtung zufällig wählt, bilden sie zwangsläufig einen erkennbaren Kreis, wenn die Anzahl der Ameisen ausreichend groß ist. Diese zwei Beispiele sollen einen Eindruck von der Vielfalt und Originalität der Ansätze, der so genannten „Programmierparadigmen“, vermitteln, die in der Informatik entwickelt worden sind. Die Beispiele sollen auch illustrieren, wie schnell mit Informatiksprachen Sprachansätze erprobt werden können. Beide Eigenschaften, die Fähigkeit zu vielen und zu schnellen Sprachexperimenten, zeichnen die Informatik aus und grenzen die Informatiksprachen von den „menschlichen Sprachen“ ab. Vermutlich ist es auch möglich, relativ schnell eine potentielle „menschliche Sprache“ künstlich zu definieren. Eine neue „menschliche Sprache“ in einer Gemeinschaft ausreichend und schnell zu verbreiten, damit sie nach kurzer Zeit verwendet wird, stellt aber eine große Herausforderung dar und ist bekanntlich selten geschehen. Im Gegensatz dazu können Informatiksprachen dank ihrer automatischen Ausführbarkeit relativ schnell und vor allem experimentell spielerisch erprobt und dadurch gelernt werden. Ist es grundsätzlich auszuschließen, dass dies in Zukunft zumindest auf eingeschränkte Aspekte „menschlicher Sprachen“ übertragen werden könnte?

Die Wahl eines Wirklichkeitsausschnitts und seine „Kodierung“ sowie die Automatisierung von Verarbeitungsprozessen verlangen Fachwissen sowohl in der Informatik als auch auf dem Anwendungsgebiet — z.B. der Verwaltung von Patientenakten. So können Informatiker als Schriftführer unserer Zeit angesehen werden. Sie sind die unvermeidbaren Akteure einer unabdingbar gewordenen Automatisierung. Es wäre müßig zu sinnen, ob die menschliche Sprache zuerst zum Ausdruck von Gefühlen oder zur Lösung von praktischen Problemen verwendet wurde: Zum Festhalten von Gefühlen und zum Erhalt des emotionalen Gleichgewichts haben Menschen höchstwahrscheinlich immer auf ähnliche Ansätze zurückgegriffen wie zur Lösung von sonstigen Problemen! Die Informatiker, diese neuen Schreiber unserer Zeit, haben insbesondere in den letzten zwei Jahrzehnten Ansätze weiterentwickelt und an die sprachliche zwischenmenschliche Kommunikation angepasst. Dabei haben sie sich an die ihnen geläufige (und bewährte!) Vorgehensweise gehalten: Sie haben Ausschnitte der Wirklichkeit ausgewählt, eine Darstellung der Daten, das heißt Formate oder Kodierungen, sowie deren Verwaltung festgelegt und schließlich automatisiert. Statt sich mit „Schildkröten“ und „Ameisen“ wie bei den Programmiersprachen Logo und StarLogo zu beschäftigen, haben sie dabei ihre Aufmerksamkeit auf in „menschlichen Sprachen“ geschriebene Texte gerichtet. Technische Lösungen für die Instandhaltung und Wartung von komplexen Computer-Netzen, wie etwa Kommunikationsforen und Newsgroups, oder für die Entwicklung von großen und komplexen Software-Systemen in Teamarbeit, wie etwa so genannte Programmierumgebungen und Versionsverwaltungssysteme, oder für das schnelle Wiederfinden von Daten, Programmteilen oder -erläuterungen in großen Sammlungen, haben sie auf allerlei Texte ausgeweitet. Dabei haben sie die geschriebenen „menschlichen Sprachen“ durchaus wesentlich erweitert: mit Hyperlinks, mit automatisch erstellten Indexen und nicht zuletzt mit Suchmaschinen. Kommunikationsforen und das World Wide Web haben unsere Kommunikation verändert. In einem knappen Jahrzehnt haben diese technischen Möglichkeiten unseren Bezug zu „menschlichen Sprachen“ und damit die Kultur wesentlich verändert. Es ist davon auszugehen, dass sich dadurch zumindest die geschriebenen „menschlichen Sprachen“ nach und nach verändern werden.

### **3. Grafische und Gebärdensprachen in der Informatik**

Die Sprachen der Informatik sind, wie bereits bemerkt, „linear“. Die Linearität von Sprachen ist uns so vertraut, dass wir uns ihrer selten bewusst sind. Eine lineare Ausdehnung ist aber keineswegs die einzige Möglichkeit für Schriftsprachen. In der Tat verwenden Menschen allerlei mehrdimensionale, vorwiegend zweidimensionale, „Schriften“ wie etwa Pläne, Diagramme und Bilder. Informatiker entwickeln und verwenden nicht nur eindimensionale, sondern auch mehrdimensionale Schriftsprachen, deren Bausteinen in der Regel Texte und Diagramme sind. Die meisten dieser „Diagrammsprachen“ sind zweidimensional. Es gibt aber Ansätze, die den Aufbau und die (zweidimensionale) Wiedergabe von dreidimensionalen Gegenständen ermöglichen. Ferner gibt es Informatiksprachen zur Darstellung von beweglichen, oder „animierten“, zwei- oder dreidimensionalen Gegenständen. Die Zeitachse zur Animation stellt dann eine dritte beziehungsweise vierte Dimension dar. Was die Diagrammsprachen der Informatik von den sonstigen Diagrammformalismen abhebt, sind zum einen ihre Dynamik, zum anderen ihre Interaktivität.

**Dynamik der Diagrammsprachen der Informatik.** Viele der Diagrammsprachen der Informatik ermöglichen bewegliche Diagramme. Seit dem World Wide Web, also seit den 1990-er Jahren, sind bewegliche Diagramme und Bilder weit verbreitet. Sie werden zum Beispiel oft für Werbezwecke eingesetzt. Sie finden aber auch Anwendung in der Technik, zum Beispiel bei der Entwicklung von Konstruktionsteilen oder bei der Simulation von komplexen zusammengesetzten Gegenständen. Dynamische Diagramme und Bilder haben heutzutage einen sehr breiten Nutzerkreis: Die verbreitete Präsentationssprache PowerPoint bietet viele, zugegeben einfache, Möglichkeiten zur „Dynamisierung“ von Texten, Diagrammen und Bildern. Es ist davon auszugehen, dass Büro- und Textverarbeitungssoftware nach und nach weitere in der Technik bereits eingesetzte Möglichkeiten der Animation aufgreifen wird. Damit wird sich zwangsläufig ein ganz anderer Bezug zu Schriftsprachen in der Kultur verbreiten.

**Interaktivität der Diagrammsprachen der Informatik.** Die meisten Diagrammsprachen der Informatik ermöglichen eine Interaktion mit dem Leser: Verweise (d.h. Hyperlinks) können verfolgt, Text- oder Diagrammteile können auf- und zugeklappt werden usw. Fortgeschrittene Formen der Interaktivität werden unter anderem im Bereich eLearning erprobt und eingesetzt, also um Lehrmaterialien einem Leser durch Interaktivität attraktiver und zugänglicher zu machen. Dabei werden oft ausgereifte „Benutzermodelle“ eingesetzt, die in engem Bezug zu den Lehrinhalten stehen. Diese technischen Möglichkeiten werden sich nach und nach verbreiten. Es ist davon auszugehen, dass auch interaktive Texte und Bilder einen immer breiteren Nutzerkreis haben werden und folglich die Kultur immer mehr prägen werden.

Neben den „Diagrammsprachen“ entwickeln Informatiker auch Programmiersprachen, die sich so genannter „visueller Paradigmen“<sup>10</sup> bedienen. Damit werden Programme visuell als in der Regel ziemlich abstrakte Bilder oder Diagramme dargestellt, die mehr oder weniger Textteile beinhalten. Das Verfassen eines „visuellen Programms“ erinnert an die virtuelle Gestaltung eines Wohnraumes mit einer der dafür verbreiteten Software. Man spricht von der „visuellen Programmierung“. Bisher stellt die visuelle Programmierung eine Randerscheinung der Informatikforschung und industriellen -entwicklung dar. Tatsache ist aber, dass dieser Bereich der Informatik als Labor zur Entwicklung und Erprobung von neuen visuellen oder diagrammatischen Ansätzen dient. Erweist sich ein Ansatz als nützlich, leicht zu erlernen und zu verwenden, so findet er sich bald in verbreiteten Softwaresystemen wie etwa Bürosoftware.

---

<sup>10</sup> S. (Poswig, 1996).

Neben den Diagrammsprachen und den visuellen Programmiersprachen werden in der Informatik neue Arten von Gebärdensprachen und Gestik zur Ergänzung der computergestützten gesprochenen, textuell oder diagrammatisch geschriebenen Kommunikation untersucht.<sup>11</sup> Dabei werden zwei unterschiedliche Ziele verfolgt. Zum einen wird versucht, die in der gewöhnlichen zwischenmenschlichen Kommunikation auftretende Gestik automatisch zu erfassen und richtig auszulegen.<sup>12</sup> Einige Forschungsprojekte auf dem Gebiet haben zum Ziel, Emotionen, die etwa Körper- und Gesichtsausdrücke vermitteln, zu entschlüsseln, um sie zum Beispiel bei der Steuerung von Geräten oder Software zu berücksichtigen.<sup>13</sup> Einer sichtbar gestressten Person könnte zum Beispiel zunächst nur die als besonders dringend oder wichtig eingestufte E-Mail angezeigt werden. Angsterscheinungen bei einer einzelnen Person oder bei einer Gruppe von Personen in einer U-Bahn könnten zum Beispiel automatisch erkannt werden. Zum anderen werden neuartige Gebärdensprachen entwickelt, womit Geräte gesteuert werden könnten.<sup>14</sup> Dies kann sowohl für die Arbeit unter extremen Bedingungen, etwa bei der Steuerung eines Flugzeuges, als auch für ältere oder körperlich eingeschränkte Personen von großem Nutzen sein. Es sei bemerkt, dass die Informatik dadurch einen Bereich der menschlichen Kommunikation zu unterstützen versucht, der zweifelsohne Bestandteil der menschlichen Sprachfähigkeit ist. Dies kann als ein weiterer Beleg der Nähe der Informatiksprachen zu den „menschlichen Sprachen“ angesehen werden.

Oft entstehen in Forschungsprojekten neue, möglicherweise ursprünglich nicht beabsichtigte Sprachen, wie etwa die „Ikonensprachen“ zur sehr abstrahierten Darstellung von Emotionen, die sich in der computergestützten Kommunikation (z.B. per E-Mail oder ICR, das heißt Internet Chat Relay) verbreitet hat. Die bekanntesten, wenn auch nicht die einzigen, sind die so genannten „Emotikonen“ :- ) oder :- ( usw. Solche Erscheinungen ähneln der Entstehung von Soziolekten oder Slangs. Auch diese Ähnlichkeit unterstreicht die Nähe der Informatiksprachen zu den „menschlichen Sprachen“.

Es ist zweifelsohne davon auszugehen, dass viele weitere sprachähnliche Ansätze, die nicht auf gesprochenen oder geschriebenen Worten, sondern auf statischen oder animierten Diagrammen und Bildern beruhen, erscheinen werden. Dies wirft die Frage auf, ob eine Abkehr von der traditionellen Schrift, wenn nicht überall und nicht für alle, so doch zumindest im Rahmen mancher Kommunikationsformen, stattfinden wird. Diese Frage ist keine freie Spekulation ohne praktische Folgen: Wie sie beantwortet wird, beeinflusst Forschungsziele im Bereich der „visuellen Informatiksprachen“. Eine weitere Frage ist, inwiefern Bilder oder Diagramme sowie Animation und Gebärden Einzug in die zwischenmenschliche, geschriebene oder nicht geschriebene Sprache finden können. Dass solche Ergänzungen der derzeitigen zwischenmenschlichen Kommunikation keineswegs auszuschließen sind, kann sicherlich angenommen werden. Die Ergänzung der gesprochenen Sprache mit der Schrift kann vermutlich als Vorbild dafür angesehen werden, wie computergestützte Bilder, Diagramme, Animation und Gebärden als weitere Ergänzungen in die zwischenmenschliche Kommunikation Einzug finden könnten.

---

<sup>11</sup> S. (Gratch, 2006) und (Yang and Ahuja, 2001).

<sup>12</sup> S. (Yang and Ahuja, 2001).

<sup>13</sup> S. (Gratch, 2006).

<sup>14</sup> S. (Yang and Ahuja, 2001).



## **4. Sprachen im ständigen Wandel**

Ein Forschungsartikel, der Ende der 60-er Jahre veröffentlicht wurde, hatte zum Titel: „Die nächsten 700 Programmiersprachen“.<sup>15</sup> Während der 60-er Jahre sind tatsächlich um die 700 Programmiersprachen entworfen worden und viele Informatiker waren über diese „Babel-Erscheinung“ ziemlich besorgt. Die Sprachvielfalt kann ja, in der Informatik wie in der Kultur, ein Hindernis der Kommunikation und Zusammenarbeit sein.

Heute ist in der Forschung die Entwicklung von neuen Informatiksprachen etwas in Verruf geraten. In der industriellen Praxis hat sie aber einen festen Platz, weil sie sowohl die Ergreifung von neuen Tätigkeitsfeldern erleichtert wie auch eine Position auf dem Markt befestigen kann. Eine der letzten weltweit erfolgreichen Programmiersprachen ist zum Beispiel Java.<sup>16</sup> Ihre Entwicklung und Verbreitung diente vor allem dazu, ein potentiell Monopol im Bereich der Programmierung von Web-Systemen zu verhindern. Neben der gezielten Entwicklung und Verbreitung von Informatiksprachen in der Industrie ist seit einem bis zwei Jahrzehnten auch in der so genannten Frei-Software-Szene ein Wildwuchs von Sprachen zu beobachten. Dabei handelt es sich um wenig organisierte, meist freiwillige Software-Entwickler, die in kleinen oder größeren weltweit agierenden Gemeinschaften Software entwickeln, darunter auch Sprachen.

Allerlei Kommunikationssoftware ist in der Frei-Software-Szene entwickelt worden, die zweifelsohne unseren Bezug zu „menschlichen Schriftsprachen“ verändern werden: „Blogs“ genannte Tagebücher im Web,<sup>17</sup> „NewsClouds“<sup>18</sup> zur Indizierung von Nachrichten, „semantische Annotationen“ aller Art zur inhaltlichen Ergänzung oder Erklärung von Wörtern und Ausdrücken in Texten, usw. So entstehen, wohl bemerkt weitgehend außerhalb der akademischen Forschung, neue sprachbasierte Kommunikationsformen. Sie sind in zweierlei Hinsichten sprachbasiert: Zum einen verwenden sie oft neue Programmieransätze und -sprachen, zum anderen beruhen sie auf und/oder ermöglichen sie neue Formen der menschlichen sprachbasierten Kommunikation.

Es ist davon auszugehen, dass computergestützte Formen der sprachbasierten Kommunikation sich noch viel mehr verbreiten werden als es heute der Fall ist. Es ist auch davon auszugehen, dass in sehr kurzen Abständen solche Kommunikationsformen von einer breiten Gemeinschaft erprobt und verändert werden. Die Frage stellt sich, inwieweit dies die „menschlichen Sprachen“ verändern könnte.

Eine weitere Frage, die sich stellt, ist, inwiefern die schnelle Entwicklung und Erprobung von Informatiksprachen sich auf „menschliche Sprachen“ ausdehnen könnte. Sicherlich wäre eine Voraussetzung für eine solche Ausdehnung, dass die „menschlichen Sprachen“ viel mehr als es heute der Fall ist automatisch erfasst und verarbeitet werden könnten. Ist aber eine solche Entwicklung unwahrscheinlich?

## **5. Rechnen: Eine Besonderheit der Informatiksprachen?**

Computer sind Rechenmaschinen und Informatiksprachen sind zum Rechnen geeignet, wobei „menschliche Sprachen“ nur begrenzt und nebensächlich mit Rechenarbeiten in Verbindung

---

<sup>15</sup> S. (Landin, 1966).

<sup>16</sup> S. (Sun, 2006).

<sup>17</sup> S. (Brady, 2005) und (Möller, 2006).

<sup>18</sup> S. <http://www.newsclouds.net>.

zu bringen sind. Stimmt das wirklich? „Menschliche Sprachen“ dienen vor allem dazu, Schlussfolgerungen zu ermöglichen, und Schließen ist eine Form des Rechnens. Letzteres wurde von Leibniz im 18. Jahrhundert postuliert, und diese Erkenntnis ist zur Grundlage der Logik, der Mathematik und der Informatik geworden. So betrachtet können Informatiksprachen als eine natürliche Fortsetzung der „menschlichen Sprachen“ angesehen werden. Die Tatsache, dass Informatiksprachen die Nutzung von Geräten voraussetzen, sollte nicht überbewertet werden. Auch die Schrift setzt Werkzeuge voraus, denen ihre ersten Benutzer zweifelsohne ähnlich „magische“ oder unverständliche Fähigkeiten zugesprochen haben. Somit wird die Frage gestellt, ob Rechnen eine Besonderheit von Informatiksprachen oder vielmehr die Wahrnehmung einer verborgenen Fähigkeit der menschlichen Sprachfähigkeit ist.

Was zeichnet also die Rechenfähigkeit von Informatiksprachen aus? Wie unterschiedlich sie auch sein mögen, lassen sich bisher alle Informatiksprachen auf ein einheitliches Berechnungsmodell zurückführen. In der Informatik sind ziemlich unterschiedliche Berechnungsmodelle<sup>19</sup> — Turing-Maschinen, rekursive Funktionen, von-Neumann-Modell usw. — vorgeschlagen und untersucht worden. Alle diese Modelle haben sich aber immer insofern als äquivalent erwiesen, dass sie alle dieselbe formale Ausdruckskraft besitzen, d.h. dieselben Berechnungen ermöglichen. Trotz intensiver Versuche ist es bisher weder gelungen, diesen Berechnungsrahmen zu sprengen, noch einen Grund für eine prinzipielle „Grenze der Berechenbarkeit“ mathematisch zu beweisen. Jedoch wird es unter Informatikern für durchaus möglich gehalten, dass neue, möglicherweise mächtigere Berechnungsmodelle noch erfunden werden können. Hoffnungen diesbezüglich werden in die Biologie und in die Kernphysik gesetzt.

Die Ausdruckskraft einer Informatiksprache wird in zweierlei Weise untersucht: einerseits informell-empirisch, andererseits formal-mathematisch.

Bei der informellen Untersuchung der Ausdruckskraft einer Informatiksprache geht es um die nicht näher formal erfassbare „Bequemlichkeit“ der Programmierung in einer Informatiksprache zur Daten- oder Berechnungsspezifikation. Die Eigenschaft ist aus demselben Grund nicht näher formal erfassbar, wie ein Vergleich der Eignung von „menschlichen Sprachen“ zum Ausdruck gewisser Gedanken müßig wäre. Fehlen einer „menschlichen Sprache“ notwendige spezialisierte Begriffe, so können sie immer der Sprache hinzugefügt werden, und das geschieht auch bei Bedarf. Psychologen mögen festgestellt haben, dass Kleinkinder, die in manchen Sprachräumen aufwachsen, gewisse Begriffe oder sprachliche Fertigkeiten früher lernen als Kleinkinder aus anderen Sprachräumen. Solche Unterschiede in der Reihenfolge, in der etwas gelernt wird, hat aber nicht zu dem Schluss berechtigt, dass eine „menschliche Sprache“ einer anderen überlegen wäre. Wie zu erwarten ist, sind die Vorstellungen unter Informatikern von der „Bequemlichkeit“ der Programmierung in einer Informatiksprache zur Daten- oder Berechnungsspezifikation oft emotional begründet und folglich schwer nachvollziehbar. Fest steht aber, dass manche Berechnungen oder Aufgaben sich mit manchen Programmiersprachen besonders prägnant ausdrücken lassen. Dies stellt eine weitere Ähnlichkeit der Informatiksprachen mit den „menschlichen Sprachen“ dar.

Die Ausdruckskraft von Informatiksprachen wird auch mathematisch-formal untersucht.<sup>20</sup> Dabei geht es zum einen um ihre Fähigkeit, Berechnungen auszudrücken, zum anderen um die Ressourcen (Zeit und Speicher), die sie zur Durchführung dieser Berechnungen benötigen. Die formal-mathematische Untersuchung von Programmiersprachen kann als ein in der

---

<sup>19</sup> S. (Hopcroft et al., 2001).

<sup>20</sup> S. (Hopcroft et al., 2001).

Informatik gewachsener wissenschaftlicher Zweig der Logik und der Mathematik angesehen werden. Er ist sehr lebendig und bringt immer wieder neue Erkenntnisse hervor.

## **6. Neuerung und Erweiterung der menschlichen Sprache**

Dieser Artikel stellt einen Versuch dar, einige Merkmale der Informatiksprachen einer Leserschaft von Nichtinformatikern näher zu bringen und den Umgang der Informatiker mit Informatiksprachen aller Art zu schildern. Dabei ist die Ähnlichkeit von Informatiksprachen mit „menschlichen Sprachen“ und die Ähnlichkeit des Umgangs von Informatikern mit ihren Fachsprachen zu dem gewöhnlichen Umgang von Menschen mit „menschlichen Sprachen“ unterstrichen worden. Dieser Versuch stellt zugegebenermaßen kein unbestreitbares wissenschaftliches Ergebnis dar. Vielmehr ist er als ein erster Schritt, als eine Anregung zum Nachdenken anzusehen. Die neueren Entwicklungen in der computergestützten Kommunikation ermuntern zu einer solchen Reflexion, machen sie sogar dringend nötig.

Abschließend möchte der Autor die These aufstellen, dass die Informatik als Wissenschaft und Technik zum einen auf der menschlichen Fähigkeit zur (Schrift-)Sprache grundlegend beruht, zum anderen die Sprache neu erfindet und dadurch erweitert. Welche Erweiterungen diese Neuerung hervorbringen wird und wie sie in der Kultur stattfinden wird, ist zweifelsohne eine der aufregendsten Fragen, die sich heute stellen.

### **Danksagung**

Der Autor bedankt sich herzlich bei Dr. Norbert Eisinger, vom Institut für Informatik der Ludwig-Maximilians-Universität München, für viele nützliche Hinweise und Anregungen zum Thema dieses Artikels sowie für seine Unterstützung während der Vorbereitung dieses Artikels.

### **Literatur**

M. Brady. Blogging, Personal Participation in Public Knowledge-Building on the Web. Report of the Chimera Working Number 2005-02, University of Essex, 2005. <http://www.essex.ac.uk/chimera/content/pubs/wps/CWP-2005-02-Blogging-in-the-Knowledge-Society-MB.pdf>

M. Burnett, Visual Programming. In *Webster, 1999*.

P. Ceruzzi. A History of Modern Computing. MIT Press, 1998. ISBN: 0-262-03255-4

J.-L. Chabert. A History of Algorithms – From the Pebble to the Microchip. Springer, 1999. ISBN: 3-540-63369-3

K. Dewdney. The New Turing Omnibus. Computer Science Press, 1993. ISBN: 0-7167-8271-5

G. Frege. Begriffsschrift, eine der arithmetischen nachgeildeten Formalsprache des reinen Denkens. Halle, 1879. (Eine Übersetzung ins Englische findet sich in *van Heijenoort, 1967*.)

J. Gratch. Emotions in Human-Agent Interactions. (Eingeladener Vortrag.) Proceedings of the Second Annual Artificial Intelligence and Interactive Digital Entertainment Conference, AAAI Digital Library <http://www.aaai.org/Library/library.php>, 2006.

J. van Heijenoort (Hrsg). From Frege to Gödel – A Source Book in Mathematical Logic, 1879-1931. Harvard University Press, 1967. ISBN: 0-674-32450-1

J. E. Hopcroft, R. Motwani, and J. D. Ullman. Introduction to Automata Theory, Languages, and Computation. Addison-Wesley, 2nd Edition, 2001. ISBN: 0-201-44124-1

P. A. Kidwell and P. E. Ceruzzi. Landmarks in Digital Computing. Smithsonian Institution Press, 1994. ISBN: 1-56098-311-6

F. v. Kutschera. Gottlob Frege. Walter de Gruyter, 1989. ISBN: 3-11-012129-8

P. J. Landin. The Next 700 Programming Languages. Communications of the ACM, Band 9, Seiten 157–166, 1966. <http://www.iro.umontreal.ca/~feeley/cours/ift6232/doc/the-next-700-programming-languages.pdf>

E. Möller. Die heimliche Medienrevolution – Wie Weblogs, Wiki und freie Software die Welt verändern. Heise, 2. Auflage 2006. ISBN: 3-936931-36-4

R. Moreau. Ainsi Naquit l’Informatique. Histoire des Hommes et des Techniques. Dunod, 1987. ISBN: 2-04-016962-8

M. Nagl: Softwaretechnik mit ADA 95. Vieweg, 1998. ISBN 3-528-15583-3

J. Poswig. Visuelle Programmierung – Computerprogramme auf graphischem Weg erstellen. Hanser, 1996. ISBN: 3-446-17990-9

M. Resnick. Turtles, Termites, and Traffic James. MIT Press, 1997. ISBN: 0-262-18162-2

M. L. Scott. Programming Language Pragmatics. Academic Press, 2000. ISBN: 1-55860-442-1

R. W. Sebesta. Programming Languages. Addison-Wesley, 2003. ISBN: 0-321-20458-1

D. Senftleben. Programmieren mit LOGO -- Einstieg, Praxis, Arbeitshilfen. Vogel, 1984. ISBN: 3-8023-0744-5

R. Sethi. Programming Languages – Concepts and Constructs. Addison-Wesley, 1996. ISBN: 0-201-59065-4

D. Siefkes et al. (Hrsg). Pioniere der informatik. Springer, 1999. ISBN: 3-540-64857-7  
Sun Microsystems, Inc. The Java Tutorial. 1995. <http://java.sun.com/docs/books/tutorial/>

Sun. The Java Tutorial. A Practical Guide for Programmers. 2006.

<http://java.sun.com/docs/books/tutorial/>

J. G. Webster (Hrsg). Encyclopedia of Electrical and Electronics Engineering, 24 volumes. John Wiley, 1999. ISBN: 0-471-13946-7

N. Wirth. Grundlagen und Techniken des Compilerbaus. Addison-Wesley, 1996. ISBN: 3-89319-931-4

M.-H. Yang and N. Ahuja. Face Detection and Gesture Recognition for Human-Computer Interaction. Springer, 2001. ISBN: 0792374096