

Organisches Rechnen - Revolution der Informatik

Christoph von der Malsburg

Institut für Neuroinformatik, Ruhr-Universität Bochum,
Computer Science Department, University of Southern California, Los Angeles,
Frankfurt Institute of Advanced Studies

Abstract:

Die Informationstechnologie läuft rasch in eine ernste Komplexitätsbarriere. Die Folgen sind versagende Projekte, Kosten- und Zeitüberschreitungen und fehlerhafte Funktionalität. Zur Überwindung der Krise müssen Rechner nach dem Vorbild lebendiger Strukturen zu autonomen Organismen werden. Der Mensch wird sich in Zukunft auf abstrakte Zielvorgaben beschränken, die der Rechner flexibel und autonom unter Ausnutzen von Emergenz und Selbstorganisation, Lernen und Abstraktionshierarchien in konkrete Prozesse übersetzt. Die Informatik wird in enger Kooperation mit Biologie, Physik, Mathematik etc. eine neue Kultur entwickeln müssen. Ernst zu nehmende Ansätze existieren, müssen jedoch koordiniert und verstärkt werden.

Die Informationstechnologie läuft gegenwärtig mit großer Geschwindigkeit in eine Komplexitätsbarriere. Das Mooresche Gesetz und wirtschaftliche Entwicklung vergrößern die installierte Rechenleistung von Jahr zu Jahr mit rasanter Geschwindigkeit, und die vorhandenen Systeme werden immer enger miteinander sowie mit Sensoren und Benutzern vernetzt. Gleichzeitig wird immer bessere Funktionalität erwartet — bessere Vernetzung, Flexibilität, Robustheit, Benutzerfreundlichkeit et cetera. Dies führt zu exponentiell steigender Komplexität der Systeme, eine Entwicklung, die nur aufgefangen werden könnte, wenn die Produktivität der Ingenieure und Programmierer mit gleicher Rate stiege. Obwohl diese schwer zu messen ist (Schätzungen reichen bis maximal 20%), sind sich die Experten einig, dass hier eine Schere immer weiter aufgeht. Man spricht von der Komplexitäts- oder Softwarekrise [Gib94, IBM03]. Das Ergebnis sind Kosten- und Fristen-Überschreitungen sowie Funktionsdefizite wenn nicht gar völliges Fehlschlagen von großen Projekten. Daraus erwächst wirtschaftlicher Schaden unübersehbaren Ausmaßes [NIS02]. Es erscheint völlig undenkbar, dass die nach dem Mooreschen Gesetz in der kommenden Dekade erwartete Funktionssteigerung um eine oder zwei weitere Größenordnungen mit heutigen Methoden und absehbaren Entwicklungstendenzen im Softwarebereich zu leisten ist. Sobald die Konjunktur wieder anspringt wird wieder der Ruf nach der grünen Karte laut werden, da die menschlichen Ressourcen im IT-Bereich an ihre Grenzen stoßen.

Was ist die Alternative? Im Zentrum dieser Frage steht das algorithmische Konzept, das als Turingmaschine bei der Entwicklung des Rechners Pate gestanden

hat, und das noch heute weitgehend unangefochten die Informatik und ihr Curriculum beherrscht. Dieses Konzept beruht auf einer Arbeitsteilung zwischen einem logischen Prozess in der Maschine und einem kreativen Prozess im Menschen. Der Rechner führt Instruktionen rasend schnell und mit absoluter Zuverlässigkeit aus, aber alle diese Instruktionen müssen ihm einzeln vorgegeben werden. Der Mensch allein ist im Besitz der kreativen Infrastruktur, die es braucht um Programme zu schreiben, zu entfehlern und zu interpretieren und er allein hat Zielvorstellungen, Weltwissen, Methoden und Strategien und diagnostische Fähigkeiten. Ursprünglich erschien der Rechner mit seiner Willenlosigkeit, Verlässlichkeit und Zuverlässigkeit als idealer Sklave, aber wir müssen uns fragen, ob der Mensch nicht inzwischen selbst zum Sklaven wird.

Das algorithmische Prinzip setzt keine prinzipielle Grenze, indem ja etwa der Rechnerprozess auf ein Programm selbst angewendet werden kann, um es zu modifizieren. Solange aber der Rechner nicht mit einer kritischen Masse an kreativer Infrastruktur ausgestattet ist, sind dieser Selbstmodifikation dadurch sehr enge Grenzen gesetzt, dass bei Fehlfunktion der Mensch immer wieder eingzugreifen hat und dazu der detaillierte Prozess im Rechner interpretierbar bleiben muss.

Es gibt große Anwendungsbereiche, in denen wir vom algorithmischen Prinzip nicht abweichen wollen oder können, Bereiche, in denen wir an dem genauen Ablauf des Prozesses interessiert sind. Dies gilt etwa für Finanzarithmetik oder für numerische Mathematik und die Simulation von synthetischen Modellen. Daneben sind aber inzwischen viel umfänglichere Anwendungsbereiche gewachsen, in denen uns die Details des Prozesses gleich sind, und wir uns nur für einfach nachprüfbare, von außen zugängliche Ergebnisse interessieren. Dies gilt etwa für den Datenverkehr, bei dem es uns allein auf sichere und schnelle Kommunikation ankommt. Bei solchen Prozessen ist nicht die Zuverlässigkeit des Prozesses in der Maschine wichtig, sondern die Zuverlässigkeit des an der Oberfläche beobachtbaren Verhaltens.

Dies gilt auch für die organischen Strukturen des Lebens. Einzellige Wesen etwa sind in keinem praktischen Sinne algorithmisch gesteuert: es gibt keinen separaten Programmierer, der das Verhalten im Einzelnen vorherbestimmt, und die Vorgänge in der Zelle sind nicht deterministisch. Dennoch sind lebende Zellen in ihrem Verhalten sicher so kompliziert wie unsere persönlichen Rechner und haben alle die angenehmen Eigenschaften wie Zuverlässigkeit, Flexibilität, Robustheit, Benutzerfreundlichkeit (geeignet definiert) und die Lern- und Evolutionsfähigkeit, die wir uns von unseren Rechnern wünschen. Und alles dieses gilt in noch viel höherem Maße von vielzelligen Organismen und insbesondere vom menschlichen Gehirn, dem Autor aller Rechnerstrukturen. Kann man vom Beispiel des Lebens lernen?

Der entscheidende Schritt besteht darin, die oben so genannte kreative Infrastruktur in der Maschine anzusiedeln und ihr so die Fähigkeit zu verleihen, übergeordnete Ziele zu verfolgen, Methoden zu entwickeln und einzusetzen, Fehler zu diagnostizieren und zu reparieren, auf neue Signale in neuer Weise zu reagieren, auf akute Situationen kreativ einzugehen und so weiter. Dies läuft darauf hinaus, dass aus dem Rechner ein elektronischer Organismus wird. Zwar werden solche Organismen (zumindest für eine lange Zeit) in dem Sinne algorithmisch gesteuert sein, dass sie

mit einem „Fundamental-Algorithmus“ ausgestattet sein werden, der die Organisationsprinzipien des Systems formuliert. Aber auf der Ebene spezifischer Vorgänge wird es keine nachprüfbar Programme mehr geben, sondern der Prozessablauf wird sich in nie exakt wiederholter Weise an die akute Situation anpassen.

Von Informatikern wird der Vision elektronischer Organismen erstaunlich selten entgegengehalten, dass sie nicht zu realisieren wäre, sie sagen vielmehr dass wir sie nie werden realisieren wollen: der Rechner soll uns dienen statt eigene Ziele verfolgen. Dies ist selbstverständlich, und deshalb muss es allein beim Menschen liegen, die Ziele der Systeme zu definieren. Andererseits wird sich im Organic Computing die Rolle des Menschen immer mehr auf genau dies beschränken können, dem Rechner Ziele vorzugeben. Diese zu formulieren ist kompliziert genug und macht schon heute einen großen Teil des Aufwands großer IT-Projekte aus (wobei das Versagen solcher Projekte oft auf dieser Ebene verursacht ist, so dass auch hier erheblicher Fortschrittsbedarf besteht). Die Sicherheit unserer Rechner vor fremden Übergriffen wird im übrigen auch erst dann gewährleistet werden, wenn die Rechner selbst alle ihre Abläufe auf abstrakter Ebene beobachten und zeitlich unmittelbar auf ihre Legalität prüfen können.

Wie realistisch ist nun dieses Ziel, elektronische Organismen zu realisieren, und wie kann es angestrebt werden? Es geht hier um einen sehr grundsätzlichen Paradigmenwechsel. Die Informatik ist in ihrer Denkweise und Methodik auf sehr fundamentaler Ebene auf das algorithmische Prinzip eingestellt: die Maschine ist digital und deterministisch, so dass man sie als logisches Gebilde ansehen kann, unter Absehen von ihrer physischen Natur. Organismen dagegen sind in ihrer Wirkungsweise analog und sind Rauscheinflüssen ausgesetzt, was eine ganz andere Beschreibungsweise voraussetzt, etwa durch Statistik und stochastische Differentialgleichungen. Der Prozess im Organismus wird durch Hierarchien von Regelkreisen auf Spur gehalten, Regelkreisen, die die Mechanismen der Organisation implementieren. Hier liegt kein Grund, die Digitalität unserer Rechner aufzugeben, aber wenn wir den Rechengang verselbständigen wollen, wenn wir ihn sensibel auf multiple Einflüsse reagieren lassen wollen, dann verliert er seinen Determinismus in jedwedem praktikablen Sinne und wir müssen mit ihm auf sehr andere Weise umgehen als bisher.

Der nötige Kulturumbruch wird wahrscheinlich am leichtesten im engen Kontakt mit den Disziplinen und Arbeitsrichtungen vonstatten gehen können, die sich schon jetzt mit relevanten Methoden befassen. Diese sind vielfältig und zahlreich, zum Teil auch in der Informatik angesiedelt, wenn auch an deren Peripherie. Zu nennen wären hier die Methoden des Soft Computing (neuronale Netze, Fuzzy Logic, evolutionäre Programme), autonome Agenten, Liquid und Amorphous Computing, Belief Propagation, Bayessche Systeme, Markov Modelle und andere. Insgesamt sind große Anstrengungen nötig, um diese Ansätze zu koordinieren und kräftig fortzuentwickeln und insbesondere auch die Ausbildung einer neuen Generation von Informatikern zu fördern, die den nötigen Kulturwandel tragen kann.

Ich möchte hier nur drei zentrale Aspekte benennen, die vorangetrieben werden müssen: Emergenz, Lernen und zielorientierte Steuerung. Von Emergenz spricht

man, wenn feinkörnige Elemente eines Systems die Tendenz haben, sich unter dem Einfluss einfacher, vorgegebener Wechselwirkungen so miteinander zu vernetzen, dass globalgeordnete Strukturen entstehen. Urbeispiel hierfür ist die Kristallisation, in der Moleküle durch van der Waals - Wechselwirkungen und geometrische Zwangsbedingungen regelmäßige Gitteranordnungen und makroskopische Kristallformen bilden. Im Bereich des Rechnersehens werden solche emergenten Phänomene etwa für die Objekterkennung durch Graphenanpassung genutzt [WFKvdM97].

Lernen ist der Vorgang, durch den ein System Strukturen aus seiner Umwelt aufnimmt und aus Beispielmustern Regelmäßigkeiten abstrahiert. Dies ist die zentrale Thematik von Machine Learning und neuronalen Netzen, aber es besteht Bedarf an Grundlagenforschung, da bisherige Systeme noch nicht die Fähigkeit des tierischen Gehirns haben, aus natürlicher Umgebung zu lernen. Zielorientierte Steuerung setzt voraus, dass ein System eine Hierarchie abstrakter Zielformulierungen hat sowie die Fähigkeit, diese Formulierungen auf konkrete Situationen anzuwenden. Dazu braucht es Erkennungsmechanismen und die Fähigkeit, die Zielhierarchie lernend an immer neue Situationen anzupassen. Hier ist der Bedarf an Grundlagenforschung vielleicht am allergrößten.

Das hier gezeichnete Bild ist eine Karikatur. Es gibt keinen harten Gegensatz zwischen Organic Computing und algorithmischer Steuerung. Die Informatik ist längst auf dem Wege, die detaillierte Behandlung von Einzelsituationen durch spezifische Programme zu ersetzen durch Programmierung auf abstrakterer Ebene, inspiriert durch abstraktere Ziele. Höhere Programmiersprachen, strukturiertes Programmieren und etwa Objektorientierung sind nichts anderes als wichtige Schritte in diese Richtung. Auch ist der ursprüngliche Determinismus, wichtige Voraussetzung für algorithmische Steuerung, für die überwiegende Masse menschlicher Wechselwirkungen mit dem Rechner ohnehin schon verlorengegangen und der Benutzer (und häufig auch der Programmierer!) sind gezwungen, experimentell vorzugehen und durch Versuch und Irrtum das an der Oberfläche beobachtbare Verhalten des Systems seinen Zielen anzupassen. Was fehlt ist, den letzten, konsequenten Schritt zu gehen und die Notwendigkeit aufzugeben, den Prozess im Rechner detailliert zu steuern oder zu verstehen. Dann würden Rechner nicht mehr von innen nach außen programmiert sondern von außen nach innen, ausgehend nicht von einer Befehlsfolge, die in komplizierten und schwierig zu durchschauenden und zu steuernden Weise zu globalem Verhalten führt, sondern vom direkten Einfluss des Benutzers, der auf das Globalverhalten einwirkt und es dem Rechner überlässt, die detaillierten inneren Vorgänge an diese Vorgaben anzupassen.

Welche Anwendungen bieten sich an? Benutzeroberflächen liegen hier nahe. Der Umgang mit schwer voraussehbaren Situationen verlangt Flexibilität und Robustheit, und obwohl hier bereits wichtige Fortschritte erzielt worden sind, etwa mit Konzepten wie der Desktop Metapher, besteht hier ein großer und wachsender Bedarf an Verbesserung. Elektronische Assistenten (bis hin zu Servicerobotern) sind ein weiterer Kandidat, wobei hier schon vieles an Konzepten angesammelt ist, besonders auch im Automobilbereich. Angetrieben von der Sicherheits- und Benutzerfreundlichkeitsthematik und getragen von einer großen Enthusiastengemeinde

und einer wichtigen Industrie könnten Rechnerbetriebssysteme zu einer Domäne des Organic Computing werden, ebenso wie Enterprise Software und (Daten-) Verkehrssteuerung.

Sehen in natürlicher Umgebung kann wahrscheinlich mit klassischer Softwaretechnik nie realisiert werden. Ein Drittel unseres Gehirns, für das man eine Rechenleistung von 10^{15} Operationen pro Sekunde schätzt, beschäftigt sich mit Sehen, und Kinder brauchen etwa 10 Jahre, bis ihre Sehleistung voll ausgebildet ist, d.h., bis sie genügend visuelle Erfahrung gesammelt haben: der Prozess des Sehens ist komplizierter, als alle bisher geschaffenen informationstechnischen Systeme. Nur durch hocheffiziente Methoden auf der Basis von Lernen und Benutzeranleitung wird breites technisches Sehen je realisiert werden. In meinem Labor und vielen anderen ist in den letzten Jahren wesentlicher Fortschritt im Sinne einer solchen Methodik gemacht worden, und der Tag ist nicht fern, dass Rechner anfangen, weitgehend autonom visuelle Erfahrung zu sammeln und auf unsere visuelle Umwelt funktionell sinnvoll zu reagieren.

In der Rückschau werden wir viele jetzt schon existente Rechnerstrukturen als Beispiele für Organic Computing verstehen. Was not tut ist entschlossene Koordination der relevanten Aktivitäten, ihre Ermutigung, sich als die Front einer neuen technischen Ära zu sehen, die Einrichtung von Curricula, die vorhandene Konzepte an eine neue Generation von Forschern weitergeben und, zumindest für eine Weile, ein intensiver interdisziplinärer Ideenaustausch mit den Lebenswissenschaften. Hier kann der alte Kontinent eine zentrale Rolle spielen. Durch ein neues Computing-Paradigma werden die Karten neu gemischt, und die Stärke Europas, kulturelle Vielfalt, kann in einer Technologie voll zum Tragen kommen, die es nicht mehr nötig hat, sich an enge Standards zu halten. Für eine junge Generation enthusiastischer Menschen gib es hier ein Eldorado zu entdecken und zu erobern.

Literatur

- [Gib94] W.W. Gibbs. Softwares Chronic Crisis. Scientific American, September:72–81, 1994.
- [IBM03] 2003. IBM’s initiative on autonomic computing, research-web.watson.ibm.com/autonomic/manifesto.
- [NIS02] 2002. NIST Study (2002), www.nist.gov/director/prog-ofc/report02-3.pdf.
- [WFKvdM97] Laurenz Wiskott, Jean-Marc Fellous, Norbert Krüger, and Christoph von der Malsburg. Face Recognition by Elastic Bunch Graph Matching. IEEE Transactions on Pattern Analysis and Machine, Intelligence, 19(7):775–779, 1997.