



Title: Gesichtspunkte zur sprachlichen Formulierung in
Vielfachzugriffssystemen unter Berücksichtigung des
"Plankalküls"
Author(s): Konrad Zuse
Date: 1960
Published by: Konrad Zuse Internet Archive
Source: Document - ZIA ID: 0102

The Konrad Zuse Internet Archive preserves and offers free access to the digitized original documents of Konrad Zuse's private papers and to other related sources.

The Konrad Zuse Internet Archive is a nonprofit service that helps scholars, researchers, students and other interested parties discover, use and build upon a wide range of content in a digital archive. For more information about the Konrad Zuse Internet Archive, please contact zusearchive@zib.de.

Your use of the Konrad Zuse Internet Archive indicates your acceptance of the Terms & Conditions of Use (<http://zuse.zib.de/tou>) including the following license agreement. If you do not accept the Terms & Conditions of Use you are not permitted to use the material.

This work by Konrad Zuse Internet Archive is licensed under a
Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License
(<http://creativecommons.org/licenses/by-nc-sa/3.0/>).
Based on a work at <http://zuse.zib.de>



Attribution (BY) - You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work). Attribute with "Konrad Zuse Internet Archive (<http://zuse.zib.de>)".

Noncommercial (NC) - You may not use this work for commercial purposes.

Share Alike (SA) - If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

The usage of this document requires the consideration of possible third party copyrights, and might necessitate obtaining the consent of the copyright holder. The Konrad Zuse Internet Archive assumes no liability with respect to the rights of third parties. The Konrad Zuse Internet Archive is not responsible for the claims of any third party resulting from any infringement of copyright laws.

Gesichtspunkte zur sprachlichen Formulierung in Vielfachzugriffssystemen unter Berücksichtigung des „Plankalküls“*

Von K. Zuse, Bad Hersfeld

Zusammenfassung

Vielfachzugriffssysteme erfordern Vereinbarungen über die sprachliche Formulierung der Informationsübertragungen sowohl zwischen den Benutzern und dem System als auch innerhalb der einzelnen Teile des Systems. Das eine ist durch den „Dialog-Betrieb“ gekennzeichnet, das andere durch das Zusammenspiel rein technischer Einrichtungen. In jedem Falle sind möglichst universelle Sprachen erforderlich, die von vornherein der Tatsache Rechnung tragen, daß die organisatorischen und logischen Operationen gegenüber der reinen Zahlenrechnung bereits heute und noch mehr in Zukunft überragende Bedeutung bekommen. Die seinerzeit im „Plankalkül“ vom Vortragenden entwickelten Gedanken können hierbei vorteilhaft verwendet werden.

Wir haben durch den hervorragenden Beitrag von Herren Professor *Weizenbaum* einen guten Einblick in ein bereits gut funktionierendes Teilnehmer-Rechensystem bekommen. Einige allgemeine Bemerkungen seien mir zuvor erlaubt:

Man sollte nicht vergessen, daß MIT als eine Hochburg für Computer und Computer-Entwicklungen angesehen werden kann. Neben dem MAC-System befindet sich dort eine Anzahl größerer und kleinerer Rechanlagen. Wer also so umfangreiche Probleme hat, daß er damit allein eine Anlage auslasten kann, wird nicht das MAC-System blockieren, sondern unabhängig davon eine der anderen Anlagen benutzen. Dieser Gesichtspunkt erscheint sehr wesentlich, da man in Deutschland oft anzunehmen geneigt ist, daß ein großes Teilnehmer-Rechensystem an einer Universität alle Ansprüche befriedigen kann. Aus der Sicht der Computer-Industrie gesehen bedeutet dies, daß dadurch seit Jahren Verkaufsgespräche an

*ZIA 0102. ZuP 035/013. Version 1. Durchgesehen von R. Rojas, G. Wagner, L. Scharf

Hochschulen und Universitäten über kleinere und mittlere Anlagen praktisch erfolglos sind, da auf die in Kürze zu erwartende große „time-sharing-Anlage“ hingewiesen wird. Auch wird die Computer-Industrie von dieser Teilnehmersystem-Welle gerade in dem Augenblick betroffen wo durch die gewaltigen Anforderungen an Software die Programmierungskapazität bis an die Grenze der menschlichen Leistungsfähigkeit und die Finanzkraft der Firmen beansprucht wird. Zusätzlich zu diesen gewaltigen Aufgaben kommen nun die sehr komplizierten Software-Probleme der Teilnehmersysteme, die nur von Spitzenkräften der Programmierungskunst gelöst werden können.

Es ist daher zu verstehen, daß nach einer anfänglichen Begeisterung eine gewisse Ernüchterung eingesetzt hat. Aufgabe der Zukunft ist es, uns auf gesunde und vernünftige Lösungen einzuspielen. Unter diesem Gesichtspunkt erscheint es sinnvoll, den Blick etwas weiter in die Zukunft zu richten. Die Teilnehmersysteme entstanden zunächst unter dem Gesichtspunkt der Ökonomie, nämlich der besseren Ausnutzung einer kostspieligen Rechenanlage durch eine große Anzahl einzelner Benutzer. Das Organisationsproblem besteht dann darin, diese Benutzer möglichst zu trennen und ihnen die Illusion zu geben, daß die Anlage ihnen allein gehört.

Darüber hinaus zeichnete sich aber bereits ein weiterer Gesichtspunkt ab: Durch die gemeinsame Benutzung der gleichen Anlage ist auch eine Zusammenarbeit zwischen verschiedenen Benutzern möglich, zunächst in erster Linie dadurch, daß ein gegenseitiger Erfahrungsaustausch durch die gegenseitige Zurverfügungstellung von Programmen und Ergebnissen durchgeführt werden kann. Hier handelt es sich also nicht darum, die Teilnehmer zu trennen, sondern sie im Gegenteil in geschickter Weise zusammenzuführen. Dies führt nun zu der weiteren Perspektive, daß die heute geplanten Teilnehmersysteme nur Vorstufen einer noch höheren Form großer Datenübertragungs- und -verarbeitungssysteme sind, die sich über ganze Länder oder gar Kontinente erstrecken und beispielsweise Regierungsstellen zur Verfügung stehen.

So gesehen, kann man dann die an den Universitäten aufzustellenden Anlagen als Schulungssysteme betrachten, an denen z.B. die Probleme der Geheimhaltung, des Speicherschutzes und der Informationsverteilung erprobt werden, um dann später in ausgereifter Form auf die höheren Systeme angewendet zu werden. Es ist zu hoffen, daß die Studenten intelligent genug sind, die Anlagen solange als kleine „devils“ zu strapazieren, bis die Systeme wirklich narren- und spionagesicher sind.

Wir haben es bei den Teilnehmersystemen sowohl mit Informationsübertragungen zwischen Mensch und Maschine als auch zwischen Maschine und Maschine zu tun. Dadurch bekommt das Sprachenproblem eine erhöhte Bedeutung, und wir sollten uns an dieser Stelle der Entwicklung der datenverarbeitenden Systeme die Frage vorlegen, ob wir mit unseren heutigen Kenntnissen und Systemen den großen auf uns zukommenden Aufgaben gewachsen sind.

Ein kurzer Seitenblick auf die Physik sei gestattet. Die Physiker sind seit jeher bestrebt, die Erscheinungen der Natur auf gewisse möglichst einfache Grundelemente, z.B. Elementarkörper (Atome) zurückzuführen. Diese ersten Bemühungen haben leider bis heute noch nicht zu einem wirklich befriedigenden Erfolg geführt.

Wir Fachleute der Datenverarbeitung dagegen verfügen über ein solches echtes Atom der Information, nämlich das Bit, aus dem sich alle höheren Informationsstrukturen aufbauen lassen. Ebenso wissen wir seit langem, daß alle Rechenoperationen auf wenige Grundoperationen mit solchen Ja-Nein-Werten zurückgeführt werden können (Konjunktion, Disjunktion, Negation). Aber machen wir von diesem uns in den Schoß gelegten Geschenk den richtigen Gebrauch? Ein kurzer Rückblick auf die Geschichte der Formelsprachen mag hierauf die Antwort geben:

Als etwa mit dem Jahre 1955 das Problem allgemeiner Programmsprachen aktuell wurde, war die Entwicklung dieser Sprachen folgenden Beschränkungen unterworfen.

1. Die zu lösenden Probleme waren vorwiegend numerischer Art.
2. Wegen der vorhandenen Ein- und Ausgabegeräte mußten die Sprachen im Fernschreibcode darstellbar sein.
3. Die Sprachen mußten den Möglichkeiten vorhandener Maschinen angepaßt sein.
4. Es mußten Compiler gebaut werden. Dies ist ein besonders wichtiger Punkt, da die beste Formelsprache ohne Compiler wertlos ist. Gerade für die Compiler gilt aber besonders der Punkt 3.
5. Es wurde eine internationale Einigung angestrebt.

Unter diesen Verhältnissen entstanden bekannte Formelsprachen, wie z.B. FORTRAN und ALGOL. Angesichts der oben angeführten Beschränkungen stellten diese hervorragende Leistungen dar und konnten sich in den folgenden Jahren weitgehend durchsetzen.

Demgegenüber bestand 1945 für den Vortragenden eine grundsätzlich andere Situation. Es existierte damals in Deutschland nur das aus Berlin gerettete Gerät Z 4, welches jedoch nicht in Betrieb genommen werden konnte. Die Entwicklung des „Plankalküls“ erfolgte als rein theoretische Arbeit am grünen Tisch und war damit auch keiner der oben bereits erwähnten Beschränkungen unterworfen. Das Wort „Plankalkül“ wurde von „Rechenplan“ abgeleitet, da das Wort Programm erst später aus dem Amerikanischen übernommen wurde. Selbstverständlich hat

der „Plankalkül“ nichts mit dem heutigen „Plan Calcul“ der französischen Regierung zu tun. Die Zielsetzung war dabei von vornherein auf eine wirklich universelle, allgemein arithmetische Formelsprache gerichtet, die von folgender Definition des allgemeinen Rechnens ausgeht. „Rechnen heißt, aus gegebenen Angaben nach einer Vorschrift neue Angaben bilden“. (Anstelle des damals noch nicht in der Datenverarbeitung üblichen Ausdrucks „Information“ wurde das Wort „Angabe“ gewählt.)

Der Plankalkül geht nun ganz bewußt vom Ja-Nein-Wert als Grundelement der Information aus und baut zunächst einen Formalismus zur Kennzeichnung komplizierter Strukturen auf, von denen einige als Beispiel angeführt seien:

$S0$	Ja-Nein-Wert (Bit)
$S1.n$	Folge von n Bit oder ganze n -stellige Binärzahl
$m1.n$	Liste von m Binärzahlen (Vektor)
σ	Angabe unbestimmter Struktur
2σ	Paar
$m2\sigma$	Paarliste

Die Kennzeichnung einzelner Variablen erfolgt der Übersicht halber in Zeilendarstellung. Die oberste Zeile enthält die Formel in üblicher Schreibweise, die zweite Zeile enthält die Variablenindizes. In der dritten Zeile stehen die Komponentenindizes, mit denen einzelne Teile aus den Informationsstrukturen herausgegriffen werden können. Die letzte Zeile schließlich enthält die Strukturindizes.

Beispiel:

V	v	v	v	v
K	3	3	3	3
S	$m2 \times 1.n$	$2 \times 1.n$	$1.n$	0
	Paarliste	i – tes Paar	Vorderglied des i – ten Paares	Bit Nr. 7 des Vordergliedes des i – ten Paares

Auf diese Weise ist es möglich, an jede einzelne Unterstruktur einer gegebenen Information heranzukommen und diese als Variable in die Rechnung einzuführen. Die Strukturzeile enthält an sich redundante Information, da die Strukturen sich aus der Art der Verknüpfung ergeben. Jedoch dient die Kennzeichnung der Struktur wesentlich der Arbeitserleichterung.

Auf die eigentlichen Rechnungsverknüpfungen kann hier der Kürze halber nicht weiter eingegangen werden. Im Plankalkül ist das Ergibtzeichen \Rightarrow und das Zeichen \rightarrow für bedingte Planteile eingeführt. Ferner sind sämtliche Möglichkeiten der Adressenvariationen und rekursiver Formeln (dort als Wiederholungsplan bezeichnet) behandelt.

Wesentlich beim Plankalkül ist auch noch die bewußte und offene Einführung der logischen Operationen. Begriffe und Kalküle der Mengenlehre des Aussagenkalküls, des Prädikaten- und Relationenkalküls werden auf eine rechnerische Form gebracht. Z.B.:

(x)	$(x \in V_0 \rightarrow R(x))$	All-Operator
(Ex)	$(x \in V_0 \wedge R(x))$	Existenz-Operator
(\acute{x})	$(x \in V_0 \wedge R(x))$	„Derjenige ...“
(\hat{x})	$(x \in V_0 \wedge R(x))$	„Diejenigen ...“
(μx)	$(x \in V_0 \wedge R(x))$	„Das nächste ...“
$(N(x))$	$(x \in V_0 \wedge R(x))$	„Anzahl von ...“

Der so entstandene Plankalkül wurde an einer Reihe von Anwendungsbeispielen erprobt. Es wurden folgende Gebiete bearbeitet:

1. Programme für allgemeine Strukturen.

Es handelt sich hier darum, unabhängig von der Anwendung auf spezielle Probleme ein Grundgerüst von Programmen zu erstellen, welches mit verschiedenen häufig wiederkehrenden Grundstrukturen arbeitet; so z.B. der Listenkalkül, die Behandlung von Paarlisten zur Darstellung von Relationen usw. Ebenso wie in der numerischen Mathematik beispielsweise der Matrizenkalkül als solcher unabhängig von spezieller Anwendung entwickelt wurde, ist es vorteilhaft, einen Satz von Grundprogrammen für Strukturen allgemeiner Art zu haben.

2. Darstellung arithmetischer Operationen.

Es handelt sich darum, die oft recht komplizierten Strukturen der in Rechenanlagen durchgeführten arithmetischen Operationen (z.B. Gleitkommaoperationen in einem gegebenen Code) bis ins letzte Bit exakt zu formulieren (also das, was man heute als Mikroprogrammierung bezeichnen würde).

3. Symbolische Rechnungen.

Hierbei handelt es sich darum, die programmtechnischen Voraussetzungen für das Operieren mit Zeichenfolgen zu schaffen. Diese symbolischen Rechnungen sind inzwischen an anderer Stelle intensiv behandelt worden, und es braucht auf ihre Bedeutung daher nicht weiter hingewiesen zu werden.

4. Behandlung von Aufgaben des Schachspiels.

Dieses Gebiet wurde gewählt, weil es auf kleinem Raum eine Fülle von Kombinationsmöglichkeiten bietet, die mit Hilfe der erwähnten beliebig aufbaubaren Strukturen und logischen Operationen gut erfaßt werden können. Es handelt sich dabei nicht darum, etwa im Sinne der Spieltheorie ein gutes Schachspielprogramm zu entwickeln, sondern die Leistungsfähigkeit des Plankalküls an einem hierfür besonders geeigneten Beispiel zu erproben. Es wurde ein Programm entwickelt, welches an Hand einer in üblicher Form

durch Angabe der einzelnen Züge dargestellten Schachpartie festzustellen hat, ob diese Partie den Spielregeln entsprechend erfolgte.

Ein sehr wichtiges Gebiet, nämlich das der numerischen Rechnungen, wurde nicht behandelt. Zunächst erschien es zu trivial als Erprobungsmittel für den Plankalkül im Vergleich zu den komplizierten anderen angeführten Gebieten. Selbstverständlich wäre es erforderlich gewesen, nach Aufstellung und Erprobung des Plankalküls gerade dieses Gebiet besonders breit zu behandeln, da es - wie wir sahen - in den ersten Jahren des Einsatzes der programmgesteuerten Rechenmaschine besondere Aktualität hatte. Die praktischen Arbeiten an der später wieder möglichen Inbetriebnahme des geretteten Gerätes Z 4 verhinderten dann aber weitere Untersuchungen, so daß bis 1955 nichts Wesentliches geschah. Wie wir sahen, wurde in dieser Zeit das Problem der Formelsprachen unter völlig anderen Gesichtspunkten von anderen Stellen wieder aufgegriffen.

Wir stehen heute wieder in einer neuen Situation und wollen fragen, ob die oben erwähnten Beschränkungen heute noch gelten.

1. Wir sind heute längst so weit, daß die numerischen Rechnungen nur noch eine untergeordnete Rolle spielen. Im Vordergrund stehen organisatorische und logische Probleme. Es ist bekannt, daß aus diesem Grunde die für numerische Rechnungen geschaffenen und eingeführten Formelsprachen an der Grenze ihrer Möglichkeiten angekommen sind.
2. Der Fernschreibcode hat zwar auch heute noch erhebliche Bedeutung, aber wir sollten z.B. durch die inzwischen geschaffenen Geräte für 8-Kanal-Codes mehr und mehr von diesem Engpaß freikommen.
3. Wir verfügen heute über wesentlich leistungsfähigere Geräte. Insbesondere gute logische Möglichkeiten der Geräte gehören heute zur selbstverständlichen Forderung marktfähiger Computer.
4. In bezug auf den Bau von Compilern haben wir nunmehr 10 Jahre Erfahrung hinter uns. Die leistungsfähigeren Geräte sollten uns auch erlauben, den Funktionsbereich der Compiler zu erweitern.
5. Der Gesichtspunkt der internationalen Einigung hat selbstverständlich auch heute noch große Bedeutung. Jedoch liegen z.B. in der Bundesrepublik als dem wohl zweitgrößten Rechenanlagenverbraucher der Welt so umfangreiche geschlossene Aufgaben vor, daß die internationale Einigung wohl erstrebenswert ist, aber keine *conditio sine qua non* bedeutet.

Schließlich haben wir heute wesentlich mehr Geldmittel zur Verfügung als vor 10 Jahren, was den Beteiligten auch eine größere Verantwortung bei der Lösung der Problem auferlegt.

Was haben nun Teilnehmersysteme damit zu tun? Wesentlich ist, wie bereits gesagt, daß durch diese Systeme und insbesondere ihre bereits angedeuteten erweiterten Möglichkeiten ein neues Niveau der Datenverarbeitung entsteht. Der Plankalkül erscheint wegen seiner Universalität hervorragend geeignet, als Bezugssprache zwischen verschiedenen Systemen zu dienen. Selbstverständlich müßte eine Überarbeitung und eventuell eine Anpassung an eingeführte und bewährte Sprachsysteme erfolgen. Dies gilt insbesondere für den Verkehr von Maschine zu Maschine.

Auch für den Verkehr zwischen Mensch und Maschine eröffnen sich neue Perspektiven. Wir sollten dabei aus der Vergangenheit eins lernen: Lösungen, die aus der Augenblickssituation heraus geschaffen werden, erreichen auf Grund der rapiden Weiterentwicklung der Datenverarbeitung bald die Grenzen ihrer Möglichkeiten. Etwas weitsichtigere und großzügigere Planung, in disziplinierter Form durchgeführt, könnte dagegen zu Lösungen führen, die auch längeren Bestand haben. Zumindest ist es in der heutigen Situation erforderlich, grundsätzliche Untersuchungen für universelle und sowohl dem Menschen als auch der Maschine angepaßte Sprachsysteme durchzuführen. Ob eine Art Esperanto hierfür die geeignete Form darstellt, müßte durch solche Voruntersuchungen geklärt werden. Sicher sind die vor uns liegenden Probleme sehr schwierig. Wir müssen aber gegebenenfalls auch den Mut haben, uns von eingefahrenen Gleisen zu trennen, wenn diese ihre Grenzen erreichen und neue Wege zu gehen.

Vielleicht verstehen wir heute soviel von Formelsprachen, daß wir nicht mehr in der Lage sind, etwas Einfachen und Vernünftiges zu tun. Ich hoffe jedoch trotzdem, daß dies möglich sein wird.

Dieser letzte Absatz steht nur im vergrößerten Manuskript am Schluß