

Siegfried Wendt

Der Kommunikationsansatz in der Software-Technik

*data report
17(4) 1982*

Es ist allgemein bekannt und wird in Diskussionen, die softwaretechnische Probleme betreffen, immer wieder ausgesprochen, daß auf vielen technischen und wissenschaftlichen Gebieten entscheidende Fortschritte erst nach der Entwicklung adäquater und übersichtlicher Darstellungsformen möglich wurden und daß in der Software-Technik die gleiche Abhängigkeit zwischen der Entwicklung der Darstellungsverfahren und entscheidenden technischen Fortschritten besteht. Diese Erkenntnis ist Ausgangspunkt des Kommunikationsansatzes, der im vorliegenden Beitrag weiterverfolgt wird.

Einordnung des Kommunikationsansatzes

Zur Komplexitätsbeherrschung in der Software-Technik werden seit einem Jahrzehnt große Anstrengungen unternommen. Die hierzu geleisteten Arbeiten lassen sich wie folgt klassifizieren:

- Verbessern der Entwurfs- und Implementierungsverfahren durch stärkeres Einbeziehen des Computers als Analyse- und Transformationswerkzeug (Analyseansatz);
- durch Bereitstellung neuer Beschreibungstechniken zur Verbesserung der zwischenmenschlichen Kommunikation (Kommunikationsansatz);
- Verbessern der Managementtechniken.

Während hier auf die Managementtechniken nicht eingegangen wird, soll der Analyseansatz kurz kommentiert werden, dem fast alle Arbeiten der klassischen Informatik zuzuordnen sind. Dieser Ansatz entspringt dem Zweifel an der Fähigkeit des Menschen, fehlerfrei zu arbeiten. Deshalb

muß man dauernd per Computer überprüfen, ob das, was der Mensch gemacht hat, auch korrekt ist, indem man die Korrektheit streng formal beweist oder Plausibilitätskontrollen durchführen läßt. Bei diesem Ansatz müssen die Entscheidungen, die der Mensch fällt, primär für den Computer verständlich sein, d. h., zum Formulieren von Entscheidungen müssen Sprachmittel geschaffen werden, die an den Analysefähigkeiten des Computers orientiert sind. Es handelt sich hierbei bisher ausschließlich um Sprachmittel auf alphanumerischer Basis. Der ganze Bereich der Programmier- und Spezifikationssprachen gehört hierher. So war der Übergang von Assembler zu höheren Programmiersprachen sicher ein wertvoller Schritt zur Komplexitätsbeherrschung. Weitere Schlagwörter aus Arbeiten zum Analyseansatz sind »strukturiertes Programmieren«, »abstrakte Datentypen« und »Korrektheitsbeweise für Algorithmen«. Es gibt praktisch keinen bekannten Informatiker, der nicht einen Beitrag zum Analyseansatz erbracht hätte [1].

Der Kommunikationsansatz erschließt eine Dimension, die außerhalb des Analyseansatzes liegt. Der Kommunikationsansatz entspringt der Überzeugung,

daß die besonderen Fähigkeiten, in denen der Mensch dem Computer überlegen ist, nämlich Mustererkennung und intuitives Erfassen von Zusammenhängen, bei der Überprüfung von Entscheidungen aktiviert werden müssen. Dabei geht es auch um das Problem, ob das, was man sich als das Gewollte vorstellt, identisch ist mit dem, was man formal als das Gewollte hingeschrieben hat. Und dies kann natürlich kein Rechner feststellen. Aus diesem Grunde müssen beim Kommunikationsansatz die Entscheidungen, die der Mensch fällt, primär für den Menschen verständlich sein, d. h., es müssen Sprachmittel geschaffen werden, die am Menschen orientiert sind. Hier stehen Beschreibungsmittel auf grafischer Basis im Vordergrund, wobei selbstverständlich auf Text und sonstige lineare symbolische Zeichenfolgen nicht verzichtet werden kann. Der Kommunikationsansatz zielt auf die Beseitigung der Intransparenz, die dazu führt, daß sich Projekte nicht richtig planen lassen und dadurch unverhältnismäßig teuer werden. Außerdem können die gewonnenen Erfahrungen kein allgemein verfügbarer Besitz des Unternehmens werden, weil sie nur in den Köpfen einzelner existieren. Für ein solches Informationsmonopol gilt – wie für jedes Monopol –, daß es von dem, der es hat, gehütet wird, während es Außenstehende verfluchen. Erst die Verbindung von Kommunikations- und Analyseansatz kann die Software-Technik zu einer echten Ingenieurwissenschaft machen; und erst die Verbindung dieser Ingenieurwissenschaft mit angepaßten Managementtechniken kann den Durchbruch zur wirklichen Komplexitätsbeherrschung bringen.

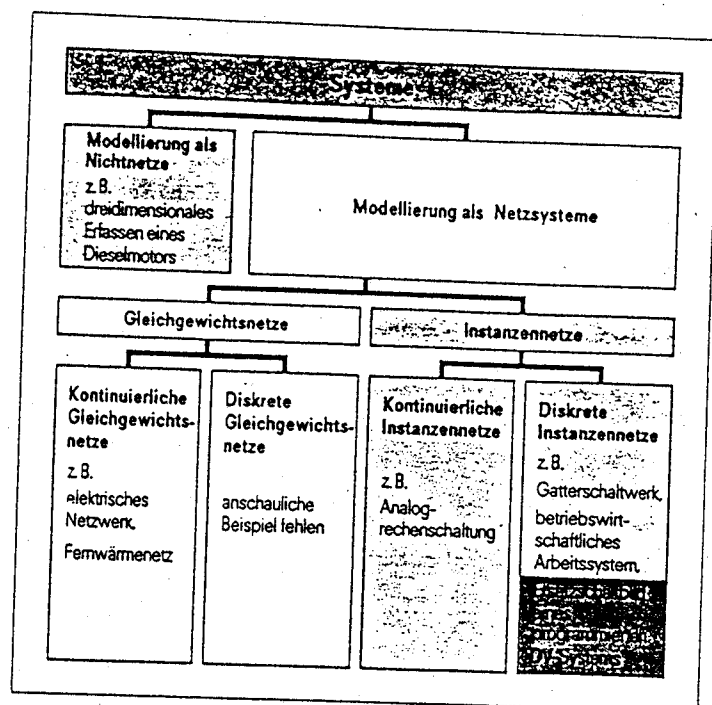
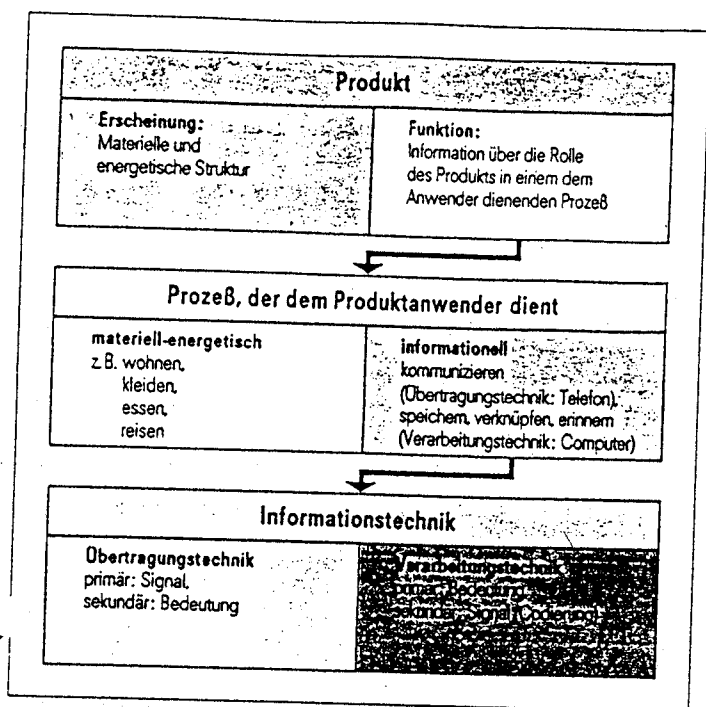


Bild 1 Zum Problem der Kommunikation über Produkte der Informationsverarbeitungstechnik

Bild 2 Klassifizierungsschema für Systemmodelle

Ursachen für die Schwierigkeiten in der Kommunikation

Anhand von Bild 1 soll untersucht werden, was an dem Produkt Software so Besonderes ist, daß das Erreichen eines hohen Kommunikationswirkungsgrads so extrem schwer ist. Die Tatsache, daß bei Software keine Serienfertigung vorliegt, kann nicht die Ursache sein, denn dieser Sachverhalt gilt z. B. auch bei Produkten, die von Architekten geplant werden. Die Ursache für die Schwierigkeit liegt vielmehr darin, daß bei Produkten, die informationellen Prozessen dienen, die Kluft zwischen Erscheinung und Funktion extrem groß ist, daß im allgemeinen nicht von der Produkterscheinung auf die Funktion geschlossen werden kann. Während von der Erscheinung einer Brücke leicht auf die Funktion geschlossen werden kann, ist es praktisch unmöglich, von der Erscheinung eines Lochkartenstapels auf die Funktion zu schließen. Es ist aber bekannt, daß in der Übertragungstechnik die Kommunikationsschwierigkeiten nicht existieren, die in der Verarbeitungstechnik die vielen Probleme mit sich bringen. Bei näherem Untersuchen stellt man fest, daß bei der Informationsübertragung das Signal im Vordergrund steht und die Bedeutung sekundär ist. Dage-

gen steht bei der Informationsverarbeitung die Bedeutung im Vordergrund und das Signal, das hier auch als Codierung bezeichnet wird, ist sekundär. Wenn man beispielsweise Telefonverbindungen baut, dann genügt es, sich über die Eigenschaften der Signale klarzuwerden; es genügt, zu wissen, daß diese Signale interpretierbar sind, aber man braucht sich über die Interpretation keine Gedanken zu machen. Für den Übertragungstechniker ist es irrelevant, was später über die Telefonverbindung gesprochen wird. Dagegen beziehen sich die Aufgabenstellungen im Bereich der Informationsverarbeitung immer primär auf Bedeutungen, während deren Zuordnung zu Signalen ein zweiter Schritt ist. Das Problem bei der Informationsverarbeitung besteht also darin, daß präzise über Bedeutungen kommuniziert werden muß. Dabei ist unter Bedeutung lediglich der Sinngehalt, das Gemeinte, zu verstehen.

Software in systemtechnischer Sicht

Hardware und Software werden meist als verschiedene Dinge auf gleicher Ebene angesehen wie Elektronik und Mechanik. Man sollte sie jedoch als zwei wesensverschiedene, aber aufeinander bezogene Dinge sehen, wie Schauspieler und Rolle. Hardware ist zum Verhalten fähig und damit Rol-

lentträger; das Verhalten ist materiell-energetischer Natur. Software ist die Beschreibung von oder die Anleitung zum Verhalten und damit Rolle; die Rolle ist informationeller Natur. Wenn Software die Verhaltensbeschreibung für Hardware ist, dann kann die Software nur dadurch beschrieben werden, daß man das durch die Software beschriebene Hardware-Verhalten noch einmal mit anderen Mitteln beschreibt und Beziehungen zwischen den beiden Beschreibungen aufzeigt. Deshalb führt die Suche nach Beschreibungsmitteln für Software zwangsläufig zu der allgemeinen Frage, wie man Systeme erfassen kann, deren Verhalten auch durch Software beschreibbar ist. Es zeigt sich, daß die hier zu betrachtenden Systeme in die Klasse der diskreten Instanzennetze eingeordnet werden können (Bild 2) [2]. Es ist zu beachten, daß die Hardware, deren Verhalten beschrieben wird, nicht unbedingt real sein muß, sondern gedacht sein darf, wie dies immer der Fall ist, wenn man zum Zweck der anschaulichen Darstellung sogenannte Ersatzschaltbilder angibt. Die FORTRAN- oder PASCAL-Maschine sind Beispiele für solche gedachten Hardware-Gebilde. Die Theorie der Netzsysteme liefert die Erkenntnis, daß zu jedem Netzsystem - und somit auch zu jedem dis-

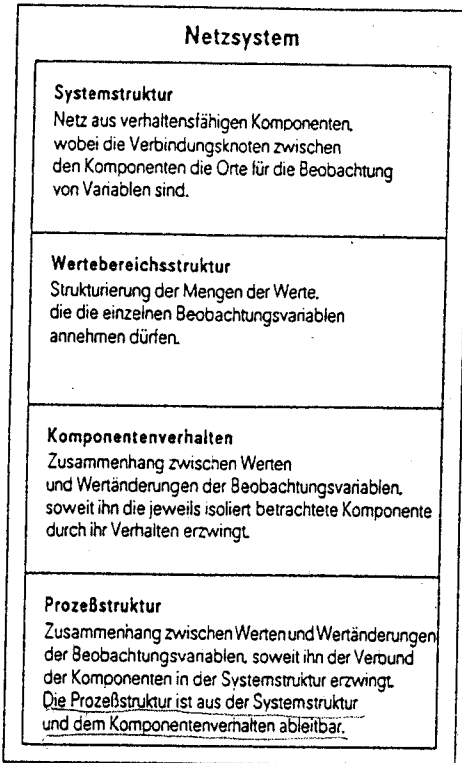


Bild 3 Die vier elementaren Aspekte eines Netzsystems

diskreten Instanzennetz – genau vier elementare Aspekte separierbar und darstellbar sind (Bild 3). Daraus wird deutlich, auf welche systemtechnischen Grundlagen man beim Kommunikationsansatz zurückgreift.

Formale Pläne

Beim Kommunikationsansatz stehen Beschreibungsmittel auf grafischer Basis im Vordergrund. Es handelt sich dabei um kommentierte formale Pläne, die nach festgelegten Regeln gestaltet und interpretiert werden. In diesem Sinn sind Landkarten natürlich formale Pläne, deren Gestaltungs- und Interpretationsregeln man bereits in der Schule lernt.

Auf der Basis der Theorie der diskreten Instanzennetze und der detaillierten Kenntnis des Aufbaus von komplexen Hardware- und Software-Systemen gelang es, Regeln zur Gestaltung und Interpretation formaler Pläne zu entwickeln, mit denen man alle relevanten Aspekte der interessierenden Systeme anschaulich erfassen kann. Neben Anschaulichkeit und eindeutiger Interpretierbarkeit mußte auch die Forderung nach hierarchischer Verfeinerbarkeit

erfüllt werden, wobei die Ergebnisse der Verfeinerung keine neuen Plan-typen erfordern dürfen. Was z. B. für die Konstruktionspläne des Maschinenbaus gilt, gilt auch hier: Die Pläne werden von wenigen qualifizierten Konstrukteuren erstellt, müssen aber von vielen Projektbeteiligten nach kurzer Schulung leicht gelesen und verstanden werden können.

Bild 4 zeigt die verschiedenen Darstellungsmittel, die bei Spezifikation, Entwurf und Implementierung von Software gebraucht werden. Die Elemente im Informationellen müssen selbstverständlich im Bereich der Implementierung letztlich bis hinunter zum letzten Bit der Codierung verfolgt werden können [3].

Die Bilder 5, 6, 7 zeigen je ein Beispiel für ein Datengitter, ein Instanzennetz und ein Kausalnetz, womit dem Leser lediglich eine Anschauung der Form vermittelt werden soll. Das Datengitter beschreibt die Struktur des informationellen Wertebereichs des Archivs im assoziativen Retrieval-System GOLEM® ohne Bezug zur Speicherstruktur. Wenn man den Auftrag geben müßte, GOLEM zu entwerfen und zu implementieren, dann wäre dieser Plan mit entsprechendem Kommentar Teil der Produktspezifikation. Beim Entwurf würde dieser Plan dann zum Ausgangspunkt der Codierung. Das Instanzennetz und das Kausalnetz gehören zum Programm AURUM, das dazu dient, eine hierarchische Struktur, die in der sogenannten Domino-Datei codiert vorliegt, mit dem Rasterplötter als Grafik auszugeben, wobei der Benutzer noch nach Parametern zur grafischen Gestaltung gefragt wird.

Bild 4 Typen formaler Pläne

Grafik	Spezifikation und Entwurf (informationelle Objekte)	Implementierung (codierte Objekte)
Systemstruktur (Komponenten und Beobachtungsorte)	Instanzennetz	Operatorenhierarchie Variablen- und Prozedurdeklarationen
Wertebereichsstruktur	Datengitter	Codierungsrahmen Typendeklarationen
Komponentenverhalten	Prädikataussagen über PRE- und POST-Werte aus dem Datengitter	Implizierte Semantik der Programmiersprachen
Prozeßstruktur (Algorithmen bzw. Protokolle)	Kausalnetz (Petrietz)	Ablaufdiagramme, Struktogramme Ablaufteil d. eigentlichen Programms

Rechnerunterstützung

Der Bereich der informationellen Strukturen und der Bereich der codierten Strukturen stehen in enger Beziehung zueinander, so daß die Pläne aus beiden Bereichen gemeinsam verwaltet werden müssen. Komplexitätsbeherrschung ist noch lange nicht gesichert, wenn die Beschreibungsverfahren bekannt sind. Komplexitätsbeherrschung erfordert sowohl eine kommunikationsfreundliche systemtechnische Begriffswelt mit der zugehörigen einheitlichen Terminologie und Symbolik als auch eine durchdachte Rechnerunterstützung für die Mitdokumentation in Form formaler Pläne und für die Abwicklung von Synthesemethoden. Mitdokumentation bedeutet das Festhalten sämtlicher Entscheidungen, und zwar möglichst in dem Augenblick, wo sie gefällt werden. Synthesemethoden sind alle Algorithmen, die irgendwie die Entwurfsentscheidungen beeinflussen, also auch Simulationsverfahren zur Entwurfsbewertung. Die Rechnerunterstützung wird selbstverständlich durch die Begriffswelt und die darauf basierenden Beschreibungsverfahren stark geprägt. Mit dem Projekt CALVADOS soll die Rechnerunterstützung für die hier vorgestellte Begriffswelt geschaffen werden.

Im Arbeitssystem, in dem sich der Software-Erstellungsprozeß abspielt, müssen die Beziehungen erfaßt sein zwischen den Entscheidungen, die ein untergeordneter Codierer fällt, wenn er eine kleine Routine programmiert, und den Entscheidungen des Chefkonstruktors, der die oberste Zerlegung des Systems festlegt. Die Rechnerunterstützung darf nicht nur darin beste-

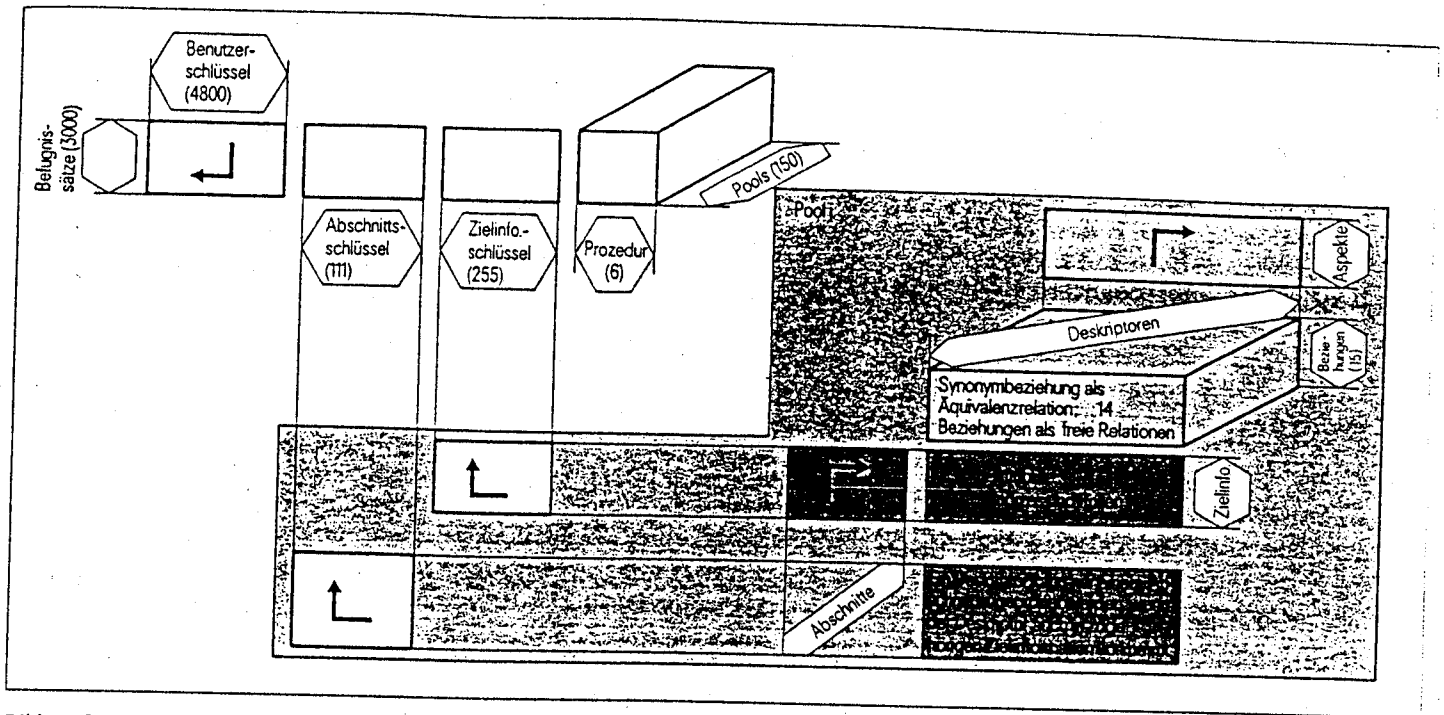


Bild 5 Datengitter des GOLEM-Archivs

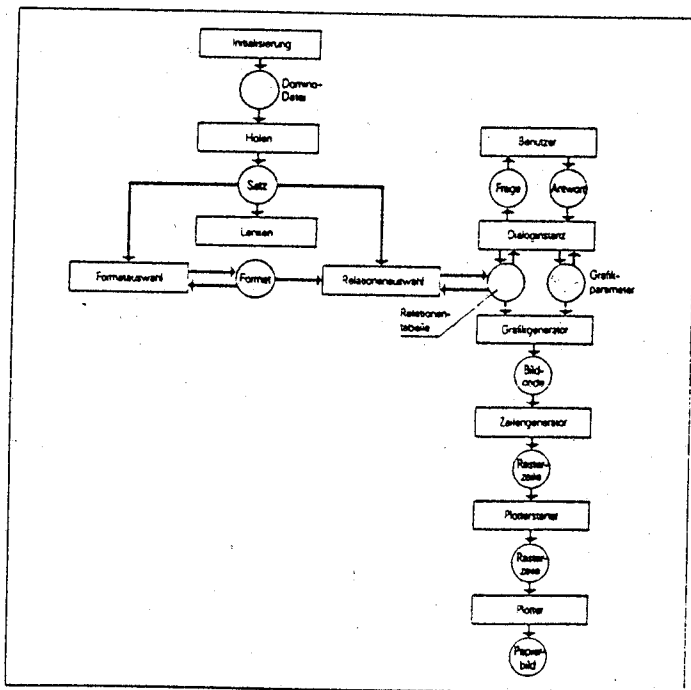


Bild 6 Instanzennetz zu AURUM

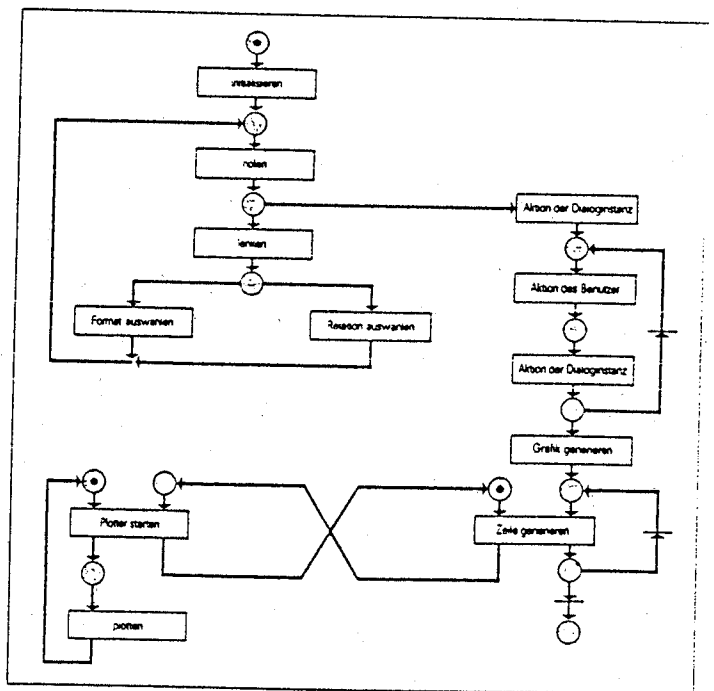


Bild 7 Kausalnetz zu AURUM

hen, Entscheidungen zu sammeln und in kommunikationsfreundlicher Darstellung auszugeben; vielmehr ist es eine zentrale Aufgabe, sämtliche einlaufenden Entscheidungen auf ihre Verträglichkeit mit bisher gefällten Entscheidungen zu überprüfen und bei Unverträglichkeit zurückzuweisen. Diese rechnerunterstützte Überprüfung der Verträglichkeit ist natürlich

nur realisierbar, wenn die Entscheidungen nach einem klaren Begriffsschema klassifizierbar sind. Ein solches Begriffsschema liegt den formalen Plänen zugrunde.

Die Entwicklung der Software-Technik zu einer soliden Ingenieurwissenschaft ist zwar noch nicht abgeschlossen, aber man ist dem Ziel deutlich näher gekommen.

Schrifttum

- [1] Wegner, P., ed.: Research Directions in Software Technology. The MIT Press, Cambridge 1979
- [2] Wendt, S.: Einführung in die Begriffswelt allgemeiner Netzsysteme. Regelungstechn. (30) 1982, S. 5 bis 12
- [3] Keller, A.: Informationelle Strukturen in Daten und ihr Einfluß auf die Programmierung. Dissert., Universität Kaiserslautern 1978