

Modellierung des VLSI-Entwurfsprozesses: Rückschau und Ausblick

Klaus Quibeldey-Cirkel, Hans Wojtkowiak
Universität Siegen, Institut für Datenverarbeitung

Zusammenfassung: Der Aufsatz untersucht die modelltheoretischen Grundlagen des Entwurfsprozesses, stellt vor und klassifiziert bisherige Modellierungsvorschläge und unterzieht sie einer vergleichenden Bewertung. Die Auswahl beschränkt sich auf Vorschläge, die Anleihen machen bei der Kybernetik, Informationstheorie und Verfahrenstechnik. Die Einordnung der untersuchten Ansätze erfolgt hierbei nach Erkenntnissen einer allgemeinen Modelltheorie. Die Studie legt offen, daß die Mehrheit der Entwurfsmodelle streng systemorientiert ist; kognitive Aspekte der Modellentwerfer und -benutzer finden kaum Berücksichtigung. Inspiriert durch den interdisziplinären Charakter der »Objektorientierung« schlagen wir eine Entwurfsmethodik vor, die kognitive Versäumnisse weitgehend vermeidet. Im Ausblick wird das Objekt-Paradigma in den Entwurfsprozeß eingeführt, werden Konzepte zur Modellierung von VLSI-Basisobjekten vorgestellt und Möglichkeiten der Konsistenzsicherung durch Vererbungsstrategien aufgezeigt.

Stichworte: Komplexität, Modelltheorie, Entwurfsprozeß, Kybernetik, Informationstheorie, Objektorientierung

EINFÜHRUNG

Das Chip-Zeitalter der Höchstintegration hat ein schwer zu überschauendes CAx-Szenario hervorgebracht, treffend umschrieben durch die Metaphern „Insel-Lösungen“ und „islands of automation“. CA-Entwurfsmethoden für applikationsspezifische integrierte Schaltungen (ASICs) und mannigfaltige CA-Techniken zur Entwurfs-synthese und -verifikation bilden ein komplexes Instrumentarium. Mit Einführung der *simulierten Netzliste* als formaler Schnittstelle zwischen Chip-Entwerfer und ASIC-Hersteller ist die Entwurfs- und Testproblematik in die alleinige Verantwortung des Hardware-Entwicklungsingenieurs übergegangen. Einerseits liegt diese Kompetenzerweiterung im Sinne des Entwicklers: als Vorzüge der vorgezogenen Netzlisten-Schnittstelle seien hier nur die verkürzte *turn-around-Zeit* und die erhöhte Entwurfssicherheit genannt. Andererseits stellt sie hohe Anforderungen an sein methodisches Systemdenken: „Welche Entwurfswerkzeuge setze ich wann und in welchem Kontext ein?“ An diese W-Fragen knüpft sich weiter die Forderung nach konsistenter Verwaltung der Datenmengen von Entwurf und Test. Also: neben *design for testability* und *design for verifiability* nun endlich auch »design for consistent data management«.

Was fehlt ist eine pragmatische Entwurfsmethodik, die seit der *Introduction to VLSI Systems* von Carver MEAD und Lynn CONWAY [1] vor einem Jahrzehnt erst in jüngster Zeit in neuen Ansätzen verfolgt wird. So hat sich zum Beispiel das neu gegründete Firmen-Konsortium CFI (CAD Framework Initiative) zum Ziel gesetzt, bislang isolierte CAD-Werkzeuge herstellerübergreifend mittels eines prozeduralen Schnittstellen-Modells zu integrieren; so stehen die massiven Modellierungs-Aktivitäten in der EDIF-Gemeinde unter dem Anspruch, den Informationsaustausch zwischen CA-Entwurfsstätten zu standardisieren; und so zeichnet sich ein neuer Trend zum *design consulting* mittels wissensbasierter Systeme ab.

Angelpunkt einer jeden pragmatischen Methodik bildet ein grundlegendes Modell des Entwurfsprozesses, das dem Modellbenutzer erlaubt, Entwurfsobjekte und -handlungen in den Gesamtkontext der Entwurfsaufgabe zu stellen. Zur Analyse und Optimierung der CAD-Einsatzmöglichkeiten, der Interaktion zwischen Mensch und Maschine und darüber hinaus zur konsistenten Datenverwaltung ist eine Studie zu den modelltheoretischen Grundlagen zwingend erforderlich. Der Aufsatz versucht, einen Beitrag in diesem Sinne zu leisten. In einem Überblick anhand der Literatur werden Vorschläge zur Modellierung des Entwurfsprozesses exemplarisch vorgestellt. Teils in Abweichung von den gewonnenen Erkenntnissen, teils unter Ausnutzung ihrer Synergie soll schließlich ein neuer Ansatz verfolgt werden: Objektorientierte Modellierungskonzepte, wie sie objektorientierten Programmiersprachen und Datenbanken gemein sind, werden eingeführt, um ein konsistentes, die kognitiven Aspekte im Systementwurf einschließendes Objekt-Modell zu entwickeln.

Die bereits angeklungene hohe Komplexität des VLSI-Entwurfs legt zunächst eine Auseinandersetzung mit der Bewältigung von Komplexität im allgemeinen nahe. Wir wollen dabei zugleich einen intuitiven Begriffsrahmen für die Grundbegriffe »Modell«, »Entwurfsaufgabe« und »Entwurfsprozeß« unserer Studie voranstellen.

1 GRUNDLAGEN

Im folgenden soll es nicht darum gehen, den Begriff der Komplexität formal zu definieren. Wir gehen von *geordneter Komplexität* im WEAVERSchen Sinne¹ aus, die sich zwar linear aus der Anzahl einfacher Systemkomponenten, aber „kombinatorisch explosiv“ aus deren Verbindungsstruktur ergibt. Hier sollen die Prinzipien der Komplexitätsbewältigung, die uns größtenteils nur intuitiv zugänglich sind, im Zusammenhang reflektiert werden.

Die „Magische Zahl Sieben“

Die experimentelle Psychologie der 50er Jahre zur Informationsverarbeitung des Menschen verstand es schon frühzeitig, Begriffe und Lehrsätze der mit dem Namen von Claude SHANNON auf engste verbundenen Informationstheorie für die einheitliche Analyse ihrer bislang verstreuten Ergebnisse zu nutzen. In seinem klassischen Aufsatz *The Magical Number Seven, Plus or Minus Two* [3] verdichtet George MILLER die wesentliche interdisziplinäre Erkenntnis:

„And finally, what about the magical number seven? What about the seven wonders of the world, the seven seas, the seven deadly sins, the seven daughters of Atlas in the Pleiades, the seven ages of man, the seven levels of hell, the seven primary colors, the seven notes on the musical scale, and the seven days of the week? What about the seven-point ra-

¹ Warren WEAVER [2] unterscheidet wissenschaftliche Aufgabenstellungen nach: „problems of simplicity“ (eine kleine Anzahl isolierbarer Variablen, z. B. in der NEWTONSchen Physik), „problems of disorganized complexity“ (eine Unzahl von Variablen mit individuell zufälligem Verhalten, deren Systemverhalten aber geordnet und statistisch bestimmbar ist, z. B. ideales Gas) und „problems of organized complexity“ (eine Vielzahl von Variablen mit individuell bestimmbar Verhalten, „... a sizable number of factors which are interrelated into an organic whole“ (S. 539)).

ting scale, the seven categories for absolute judgement, the seven objects in the span of attention, and the seven digits in the span of immediate memory? For the present I propose to withhold judgement. Perhaps there is something deep and profound behind all these sevens, something just calling out for us to discover it. But I suspect that it is only a pernicious, Pythagorean coincidence." (S. 97)

Zwar läßt er die Frage nach dem Warum — dem „design behind it“ — unbeantwortet (um die Antwort bemühen sich intensiv die Neurophysiologie und die junge Kognitionswissenschaft [4]), MILLER bringt aber das Phänomen der beschränkten Kanalkapazität des Menschen in die begriffliche Vorstellungswelt des Ingenieurs. Er rückt von der Metrik der *Varianz* als dimensionsbehafteter Größe für Information ab und interpretiert die teils noch auf dieser Quantität beruhenden empirischen Ergebnisse mit Hilfe des dimensionslosen Maßes *Bit*². Auf diese Weise gewinnt er eine klare Aussage zur kognitiven Bewußtseins-Barriere des Menschen.

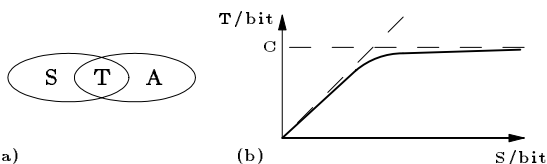


Bild 1.1: Zur menschlichen Kanalkapazität nach MILLER

Die analysierten Experimente waren methodisch gleich strukturiert. Es wurde getestet, wie genau Probanden Zahlenwerte sensorischen Stimuli zuordnen können, wobei die Antwortzeit keine Rolle spielte: „experiments on absolute judgement“. Das VENN-Diagramm im Bild 1.1a verdeutlicht die beteiligten Informationsmengen: Stimuli *S*, Antworten *A* und übertragende Information *T*. Die Anzahl der Stimuli-Alternativen wurde soweit erhöht, bis sich die übertragende Informationsmenge asymptotisch einem Grenzwert, der Kanalkapazität *C*, näherte (Bild 1.1b). Im Fall eindimensionaler Stimuli, wie beispielsweise Tonhöhe, Geschmacksintensität von Salzlösungen, Punktpositionen auf einer Linie, lag das Mittel der Kanalkapazität bei 2,6 bit mit einer Standardabweichung von 0,6 bit pro Zeiteinheit. Auch bei mehrdimensionaler sensorischer Erregung, wie Salz- und Zucker-Konzentrationen in gemeinsamer Lösung oder Punktpositionen in einer Fläche, kam MILLER auf das gleiche Ergebnis: Der Mensch vermag lediglich 7 ± 2 Kategorien („chunks of information“) gleichzeitig zu erfassen und zu verarbeiten.³

Die verblüffend geringe Kanalkapazität muß aber nicht notwendigerweise unsere kognitiven Fähigkeiten auf diese Größenordnung beschränken. Mittels *Codierung* — Gruppieren und Organisieren der Eingangs-Informationsmenge — lassen sich die „chunks of information“ semantisch beliebig mächtig erfassen und somit der Informationsengpaß aufbrechen. Daraus ergibt sich folgende

² Die SHANNONSche Kommunikationstheorie betrachtet nicht die Bedeutung, sondern ausschließlich den technischen Aspekt einer Information und ihrer Übermittlung. Als Grad der Wahlfreiheit wird *Information* definiert über den Logarithmus dualis der Anzahl der Wahlmöglichkeiten.

³ Herbert SIMON [5] verweist des Weiteren auf die begrenzte kognitive Verarbeitungsgeschwindigkeit: Im Mittel benötigt der Mensch fünf Sekunden, um einen „chunk“ ins Langzeit-Gedächtnis aufzunehmen. (S. 82)

Relevanz für die Unterstützung kognitiver Prozesse, zum Beispiel eben für die Bewältigung einer VLSI-Entwurfsaufgabe: Durch den Vorgang des Modellierens wird der Informationsgehalt eines komplexen Realitätsausschnitts derart codiert, d. h. auf wenige kognitiv gleichzeitig erfaßt und bearbeitbare Informationseinheiten abgebildet, daß die physio-psychologische Schwelle des Menschen (Kanalkapazität, Kurzzeit-Gedächtnis) scheinbar überschritten wird. Das *chunking* muß hierbei MILLERs magischer Quantität genügen. In der VLSI-Praxis bedeutet dies beispielsweise, daß die Anzahl der Abstraktionsebenen, das Grundsortiment an Funktions- und Struktur-Modul (z. B. in Zellen-Bibliotheken), die Alternativen in Benutzer-Menüs oder — allgemein gesprochen — die Menge der Klassenbildungen auf die MILLERsche Größenordnung beschränkt bleiben. Nur so kann ein Überschreiten der Bewußtseinskapazität des Modellbenutzers vermieden (der Übertrag ginge, informationstheoretisch gesehen, in *Rauschen* über), seine maximale Aufnahmefähigkeit und damit die größtmögliche Modell-Effizienz erzielt werden.

Fazit: Psychologisch stellt ein »Modell« die Codierung eines komplexen Realitätsausschnitts dar, um die kognitiven Möglichkeiten des Menschen optimal zu unterstützen.

Die „Architektur des Komplexen“

Es ist hier nicht der Raum, um auf das breite Erkenntnis-Spektrum der allgemeinen Systemtheorie einzugehen, obgleich — wie wir später noch sehen werden — systemtheoretische Begriffe und Methoden eine wichtige Rolle bei der Beschreibung des Entwurfsprozesses spielen. In diesem Abschnitt soll die wesentliche Gemeinsamkeit „erfolgreicher“ komplexer Systeme, nämlich das hierarchische Strukturierungsprinzip nach Herbert SIMON, hervorgehoben und darauf aufbauend ein allgemeines Systemmodell der Entwurfsaufgabe vorgestellt werden.

Zunächst sei motivierend die klassische *Hora-Tempus*-Parabel von SIMON aus seinem grundlegenden Aufsatz *The Architecture of Complexity* [6] skizziert:

Zwei renommierte Uhrmacher, Hora und Tempus, setzen ihre in der Komplexität von etwa 1000 Teilen vergleichbaren Uhrwerke auf unterschiedliche Weise zusammen. Während Tempus so vorgeht, daß er bei einer unvorhergesehenen Unterbrechung durch einen Kunden das bis dahin zusammengesetzte Werk gänzlich ablegen muß und dieses dadurch in seine Bestandteile wieder zerfällt, entwirft Hora seine Uhr in „ablegbare“ Einheiten. Jeweils zehn Teile bilden ein stabiles Gefüge, zehn solcher Gefüge wiederum das nächst größere und zehn dieser schließlich das Gesamtwerk. Wenn also Hora bei seiner Arbeit unterbrochen wird, so betrifft die destruktive Wirkung des Ablegens immer nur die letzte Teilkonstruktion. Auf diese Weise vollendet er sein Werk bei gleicher Unterbrechungsrate in einem Bruchteil der Arbeitszeit, die Tempus hierfür benötigt. (Beispiel: Eine Unterbrechungshäufigkeit von 1 zu 100 bewirkt im Mittel eine ca. 4000fach längere Herstellungszeit für Tempus-Uhren.)

SIMON illustriert die empirische Anwendbarkeit seiner Parabel auf eine Vielzahl biologischer, soziologischer und physikalischer Systeme. Er unterstreicht die Häufigkeit, mit der Komplexität in der Natur *hierarchisch strukturiert* auftritt. Komplexe Systeme entwickeln sich evolutionär am schnellsten aus weniger komplexen Systemen, wenn diese

jeweils *stabile Zwischenzustände* repräsentieren: „complexity evolves from simplicity“. In hierarchischen Strukturen können Interaktionen zwischen (inter) und innerhalb (intra) von Subsystemen unterschieden werden. Intra-Verbindungen sind dabei in der Regel stärker ausgeprägt als Inter-Komponentenverbindungen. Dies hat zur Folge, daß die hochfrequente Dynamik einer Hierarchie, gekennzeichnet durch „near decomposability“ (S. 69f), die innere Struktur der Komponenten betrifft, während sich die niederfrequente Dynamik auf die Interaktionen zwischen den Komponenten beschränkt. Die hierarchische Zerlegbarkeit vereinfacht somit — ganz im Sinne von MILLER — die kognitive Bewältigung von Komplexität: *divide et impera*.

Aufbauend auf dem Hierarchie-Postulat von SIMON soll ein allgemeines Systemmodell nach Günter ROPHOHL [7] beschrieben werden, das auf beliebige technische Objekte anwendbar ist. Die in der Systemtheorie gewonnenen Aspekte des Systembegriffs — funktional, strukturell und hierarchisch — verdeutlicht Bild 1.2:

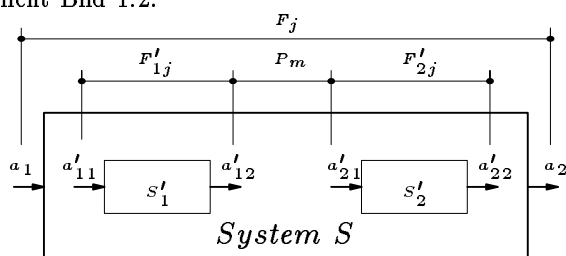


Bild 1.2: Didaktisches Modell für Sachsysteme nach ROPHOHL

Der funktionale Systemaspekt geht auf den Primitiv-Ansatz der *black box* als größtmögliche Abstraktion ihrer inneren Struktur zurück. Die Funktion F_j des Systems gegenüber seiner Umgebung steht global für die Umsetzung einer Problembeschreibung (Spezifikation/Pflichtenheft) in eine Problemlösung (Implementierung/Produkt). Die systematische Umsetzung der Ein- und Ausgangseigenschaften ist Gegenstand der Informatik schlechthin: siehe hierzu zum Beispiel Ian SOMMERVILLEs „Wasserfall-Modell“ des Software-Lebenszyklus in *Software Engineering* [8]. Der strukturelle Aspekt bezieht sich auf die Konnektivität der Subsysteme S'_i , d. h. auf das Verbindungsgeflecht aus Serien-, Parallelschaltungen und Rückkopplungen. Kurz gesagt, das Ganze ist mehr als die Summe seiner Teile: Es ist nicht trivial, von den Eigenschaften der Subsysteme auf die Eigenschaften des Gesamtsystems zu schließen. Die Struktur des Verbindungsgeflechts, gegeben durch die Menge aller Relationen P_m , bestimmt entscheidend das Verhalten des Systems. Der hierarchische Aspekt schließlich fußt auf der von SIMON aufgezeigten Notwendigkeit, Komplexität geordnet zu zerlegen. Die Komponenten S'_i bilden eigenständige Subsysteme mit möglicherweise weiterer Schachtelung bis hin zu elementaren Subsystemen, wobei SIMON mit dem Begriff der Hierarchie nicht notwendigerweise eine Rangfolge verbindet.

Fazit: Systemtheoretisch verlangt die Komplexität der »Entwurfsaufgabe« deren Dekomposition in semi-unabhängige strukturelle Einheiten (Entwurfsabschnitten) mit klar definierten Übergängen: hierarchisches Strukturierungsprinzip.

Vom Handentwurf zum Silicon-Compiler

SIMONS Hierarchie-Postulat — Komplexes entwickelt sich aus Einfachem — beweist seine Allgemeingültigkeit auch in der technischen Evolution des Entwurfsprozesses. Triebfeder jeglichen Entwerfens und Konstruierens ist der Drang des Menschen, seine reale Umwelt zu beherrschen oder seine Vorstellungen und Pläne in ihr zu realisieren. Als Werkzeugmacher und -benutzer zeichnet er sich deutlich von anderen Lebewesen aus. Psychologisch gesehen resultieren Entwurf und Einsatz technischer Mittel zur Beeinflussung seiner Außenwelt aus dem Streben des Menschen nach Organverlängerung, -unterstützung und -ersatz.

Gleichgültig, ob es sich um die Konstruktion von Software oder Hardware handelt, lassen sich mehrere Entwicklungsabschnitte unterscheiden:

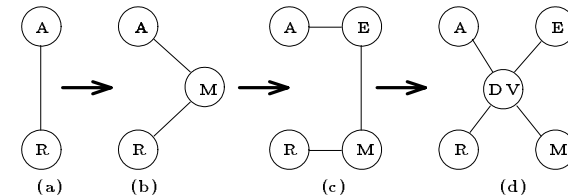


Bild 1.3: Zur technischen Evolution des Entwurfsprozesses

a: Die Situation kennzeichnet ein Zeitintervall, in dem der Realitätsausschnitt R , den es zu manipulieren gilt, noch einfach und unmittelbar zu beeinflussen ist. Ein bezüglich des Anwenders A externer Modellierungsvorgang ist nicht erforderlich. Man spricht in diesem Zusammenhang auch vom „unbewußten Modellierungsprozeß“: Entwurf und Herstellung sind eins. Die Bildhauerei ohne Vorlage möge ein Beispiel hierfür sein.

b: Mit zunehmender Komplexität der Entwurfsaufgabe stellt sich die Notwendigkeit, die zu verarbeitende Informationsmenge abschnittsweise geeignet zu modellieren. Der Modellierungsprozeß findet seine erste konkrete Ausprägung in einem externen Modell M , das die Komplexität des Entwurfsvorgangs durch Einführen von Abstraktionsebenen im MILLERSchen Sinne *codiert*. Beispiele: Nassi-Schneiderman-Diagramme, Schaltpläne.

c: Mit der Entwicklung mächtigerer Methoden und Techniken wachsen zugleich die Anforderungen und Bedürfnisse und vice versa — nimmt die zu bewältigende Informationsmenge qualitativ und quantitativ zu. Das Modellierungsproblem überfordert den gelegentlichen Anwender; es muß folglich an einen Experten oder an eine Gruppe von Experten E delegiert werden, z. B. Architekten, Programmierer und Chip-Design-Ingenieure. Mit Einführung einer weiteren Instanz in den Entwurfsprozeß mit dem Ziel der Arbeitsteilung entsteht das Problem der Kommunikation. Die Distanz zwischen Anwender und Zielobjekt vergrößert sich. Nicht immer decken sich Wunsch und

Wirklichkeit, Problembeschreibung und entworfene Lösung.

d: Eine weitere Komplexitätszunahme in der algorithmischen Verarbeitung und in der Verwaltung des Datenaufkommens überfordert schließlich selbst den Experten. Die elektronische Datenverarbeitung (DV) — metaphorisch der *Computer* — hat zwar die Effizienz der Informationsverarbeitung revolutioniert, bleibt aber in der evolutionären Entwicklungslinie des gesamten Entwurfsprozesses. Die Situation spiegelt die heutige Konstellation der beteiligten Instanzen und deren Kommunikationspfade wider. Als Beispiel stehe hier das eingangs geschilderte CAX-Szenario des VLSI-Entwurfs. Die Mensch-Maschine-Interaktion rückt in den Vordergrund. Schnittstellen-Probleme treten auf.

e: Eine derweilen noch futuristische Situation (im Bild nicht skizziert) könnte schließlich die Erfahrungen und das Wissen des Experten sowie die abstrahierte Darstellung des Entwurfsobjektes als Modell in Form von wissensbasierten Automaten integrieren. In der VLSI-Welt wäre dies der ideale „Silicon Compiler“, wie ihn Daniel GAJSKI in *Silicon Compilation* [9, S. 39f.] definiert.

Zusammenfassend können wir somit festhalten, daß der Entwurfsprozeß einen evolutionären Charakter aufweist und im SIMONSchen Sinne streng hierarchisch strukturiert ist. Die Modellbildung nimmt in allen Entwicklungsabschnitten eine zentrale Stellung ein. Zur detaillierten Geschichte des Modellbegriffs und des Denkens in Modellen sei in diesem Zusammenhang auf die Übersichtsarbeit von Roland MÜLLER [10] verwiesen.

Fazit: Die evolutionär gewachsene Komplexität des »Entwurfsprozesses« als geordnete Abfolge von Entwurfsabschnitten erfordert die koordinierte Beteiligung mehrerer Instanzen von Anwender, Experte und CAX-Instrumentarium. Aus der Sichtweise des Benutzers zielt der technologische Fortschritt auf eine Verkürzung der operativen Distanz zwischen Spezifikation und Implementierung.

2 RÜCKSCHAU

Nachdem wir den modelltheoretischen Unterbau eines allgemeinen Entwurfsprozesses erläutert haben, wenden wir uns nun eingehender den konkreten Modellierungsansätzen zu. Die gesichteten Vorschläge lassen sich grob nach dem formalen Hilfsmittel, der Meta-Sprache, ordnen, die jeweils zur Beschreibung hervorzuhebender Modellaspekte angewandt wird. Es zeigen sich vor allem Anleihen aus der Systemtheorie. Insbesondere deren Teildisziplinen — die WIENERSche Kybernetik und die SHANNONSche Informationstheorie — spielen hier die Rolle von Hilfswissenschaften, ähnlich wie die Mathematik in den Naturwissenschaften. Als dritte Modellierungsklasse haben wir (mangels eines griffigeren Terminus) den im Chemo-Produktionsbereich ange-

siedelten Begriff der Verfahrenstechnik gewählt, der sich zur Beschreibung von Verfahrensketten und -landschaften im Umfeld des VLSI-Instrumentariums zwanglos anbietet. Die Vorstellung der so klassifizierten Modellierungsansätze soll zunächst „wertfrei“ erfolgen. Eine vergleichende Bewertung werden wir im Anschluß anhand eines modelltheoretischen Kriterienkatalogs versuchen.

2.1 Kybernetische Modelle

Die Grundbegriffe, Methoden und Erkenntnisse der von Norbert WIENER begründeten Kybernetik zählen zweifellos zum Allgemeingut sowohl natur- als auch geisteswissenschaftlicher Disziplinen. Als interdisziplinäre „Wissenschaft von den Wirkungsgefügen“ (Hans SACHSSE in [11]) abstrahiert sie von der physikalischen Natur der Wirkungsträger, der Materie und Energie. Sie konzentriert sich auf die rein logisch-strukturelle Verknüpfung und erlaubt somit Analogien in Struktur und Verhalten zwischen technischen und natürlichen Systemen. Der Untertitel in WIENERS klassischem Originalwerk weist bereits darauf hin: *Cybernetics: Or Control and Communication in the Animal and the Machine* [12]. Der Universalcharakter, den die Kybernetik ihrem innewohnenden Abstraktionsprinzip verdankt — Informationsverarbeitung als das „Gemeinsame im Verschiedenen“ [11, S. 7] — prädestiniert sie als formale Meta-Sprache zur Modellierung komplexer Systeme. Was wunder, daß ein Großteil der Modellierungsansätze auf der kybernetischen Betrachtungsweise beruht.

Bevor wir auf die kybernetischen Modelle im Detail eingehen, sollen die zentralen Begriffe anhand einer allgemeinen Gleichung kurz erläutert werden. Ausgehend vom vereinfachenden funktionalen Black-box-Ansatz (siehe ENCYCLOPEDIA BRITANNICA unter *cybernetics*)

$$y_{t+k} = f(\epsilon_t) = f(x_t - \varphi_t) \quad (1)$$

lassen sich die hier interessierenden regelungstechnischen Aspekte der Kybernetik recht gut überschauen. Der *Beobachter* im Entwurfsprozeß — in unserer Diskussion der Mensch oder die Maschine als Modellbenutzer — ermittelt die Abweichung der Regelgröße x_t , des aktuellen Entwurfsergebnisses eines Entwurfsabschnitts, von der Führungsgröße φ_t , dem vorgegebenen Entwurfsauftrag. Die Differenz ϵ_t zwischen Entwurfsvorgabe und -ergebnis wird in geeigneter Form als Stellgröße auf den Eingang des Systems zurückgeführt: Prinzip der Rückkopplung. Am Systemausgang stellt sich die geregelte Größe y_t ein, verzögert um die Reaktionszeit k . Im „eingeschwungenen“ Zustand liegt schließlich der hinsichtlich der Regelabweichung ϵ_t optimierte Entwurf vor.

Der Regelkreis heißt deterministisch, wenn die Reaktion des Systems durch die Eingangsinformation bestimmt wird. Er heißt darüber hinaus dynamisch, wenn er über ein Gedächtnis für früheres Verhalten verfügt und wenn Eingangsinformation und Geschichte der zurückliegenden Einwirkungen gemeinsam die aktuelle Systemreak-

tion bestimmen. Die Identifizierung und Beschreibung der einzelnen regelungstechnischen Komponenten im deterministisch dynamischen Entwurfsprozeß sind Gegenstand der nun zu diskutierenden kybernetischen Modelle.

Kybernetisches Modell nach AMKREUTZ

Als einer der ersten, die den rechnerunterstützten Entwurf aus der kybernetischen Betrachtung analysieren, unterstreicht J. AMKREUTZ [13] den Universal-Charakter der Kybernetik als interdisziplinäre Meta-Sprache. Aus der Architektur kommend entwickelt er ein regelungstechnisches Modell des allgemeinen Entwurfsprozesses, das es erlaubt, „specific characteristics of man and machine“ (S. 187) für technische Problemlösungen zu untersuchen und zu integrieren. Vor dem Hintergrund der Evolution des Entwurfsprozesses („...the tension between more complex design requirements and objectives, and the tools offered to solve design problems“ (S. 192)) verschafft uns AMKREUTZ Zugang zur inneren Struktur der Black-box »Entwurf«. Er begreift den Vorgang des Entwerfens als informationsverarbeitenden Prozeß. Seine Vorgehensweise der „model reticulation“ (S. 188, übertragen: vernetzte Modellverfeinerung) führt auf ein Regelsystem, wie es Bild 2.1 zeigt:

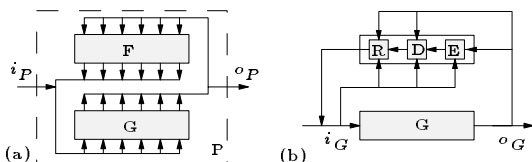


Bild 2.1: Der Entwurfsprozeß als Regelsystem nach AMKREUTZ

AMKREUTZ unterscheidet prinzipiell zwischen zwei informationsverarbeitenden Funktionsblöcken: „generation“ G und „feedback“ F . Die erzeugenden Funktionen setzen einen Entwurfsauftrag in eine Menge von Entwurfsalternativen um, die durch die rückkoppelnden Funktionen unter dem Gesichtspunkt der Realisierbarkeit und vorgegebener Optimierungskriterien (Entwurfsgüte, Kostenfunktionen) auf ein Entwurfsergebnis eingeschränkt werden. Da der Entwurf eine Vielzahl synthetisierender und verifizierender Aspekte beinhaltet, die abschnittsweise auf den Prozeß einwirken und wechselseitig alle Entwurfsabschnitte umfassen, setzt sich der gesamte Entwurfsprozeß letztendlich aus einem System gekoppelter Regelkreise zusammen (Bild 2.1a). Die Bearbeitung der einzelnen Entwurfsaspekte erfolgt im allgemeinen nicht sequentiell, so daß in der Verbindungsstruktur zwischen den generierenden und rückführenden Funktionen kombinatorische Vielfalt herrscht.

Von zentraler Bedeutung für die Beurteilung des Einsatzes und der Entwicklung von CAD-Komponenten sind die 'kybernetischen Basisfunktionen'. Funktionell teilt sich die Rückkopplung auf in: „evaluation“ E , „decision“ D und „regulation“ R (Bild 2.1b). Der Soll-Ist-Vergleich E liefert die Regelabweichung ϵ_t . Die Entscheidungsinstanz D ermittelt für diese Regelabweichung

eine geeignete Strategie und Stellgröße für den Steuerungsmechanismus. Die Steuerung⁴ R führt die gewählte Entscheidungsstrategie aus, indem sie den Eingangsinformationsfluß zielgerichtet ändert. Zusammen konstituieren die kybernetischen Basisfunktionen den Regelungsanteil von $f(\epsilon_t)$.

Der Rückkopplungszweig dient somit der Anpassung an das globale Entwurfsziel, das erst im Gleichgewichtszustand des Regelsystems angenähert vorliegt. Da jeder kybernetische Prozeßschritt die Historie vorhergehender Schritte berücksichtigt (dynamisches System), stellt der Entwurf insgesamt einen Lernprozeß dar:

„The cybernetic model developed is a dynamic decision model, in which the process is directed towards the production of an optimum design by making successive series of decisions on the basis of endogenous and exogenous information.“ (S. 184)

Für die Analyse der Einsatzmöglichkeiten von Automaten, ist die 'Quantifizierbarkeit' der kybernetischen Basisfunktionen maßgebend. Lassen sich Bewertungs-, Entscheidungs- und Steuerungsinstanzen auf formale Systeme (Algorithmen) abbilden, so sind sie auch automatisierbar. Die Zuordnung der Aufgaben an Anwender und CAD-System wird bestimmt durch die Realisierbarkeit quantifizierbarer und nicht-quantifizierbarer (im allgemeinen heuristischer) Prozeßschritte. Um eine erfolgreiche Zusammenarbeit zu gewährleisten, ist eine sorgfältige Aufgabenteilung Voraussetzung. AMKREUTZ leitet hierfür drei wesentliche Forderungen ab:

- Anpassung des CAD-Systems an die Arbeitsweise des Anwenders, um dessen Kreativität zu fördern: *Ergonomie*
- geeignete Werkzeuge für die Kommunikation: *Graphik*
- Anpassung des Anwenders an die Systematik des CAD-Systems: *Systemdenken*

Mehr-Ebenen-Erweiterung nach RAMMIG

Das abstrakte Grundmodell in Franz RAMMIGs „durchgängiger Systematik des Hardware-Entwurfs“ [14, S. 5] stützt sich auf die kybernetischen Erkenntnisse von AMKREUTZ. Mit einem Jahrzehnt technologischen Abstand verallgemeinert RAMMIG den Rückkopplungsprozeß »Entwurf« über die Abstraktionsebenen hinweg. Sein „makroskopisches Modell des Entwurfsprozesses“ ist identisch mit dem Vorschlag von AMKREUTZ. Auch er unterscheidet zwei Klassen von Entwurfs-handlungen: „generierende und überprüfende Aktivitäten“; erhöht aber die Transparenz der Informationskanäle und des Aktivitätsflusses, indem er diese genauer spezifiziert⁵.

RAMMIGs „mikroskopische“ Verallgemeinerung des kybernetischen Ansatzes führt auf ein Klassifikations-Schema, das die erzeugenden und überprüfenden Entwurfshandlungen weiter differenziert

⁴ AMKREUTZ unterscheidet hier nicht streng zwischen Steuern und Regeln, obwohl nur letzteres eine selbständige Verhaltensänderung durch Beobachten der Regelgröße impliziert.

⁵ Seine Systematik steht unter dem Anspruch, die höheren Abstraktionsebenen des Hardware-Entwurfs — System-, Verhaltens-, Registertransfer- und Gatterebene — einheitlich zu beschreiben: *Breitband-Hardware-Beschreibungssprache DACAPO III*.

in ebenen-variente — „Implementation“ und „Regulation“ (Bild 2.2) — und ebenen-invariante — „Modifizierung“ und „Optimierung“.

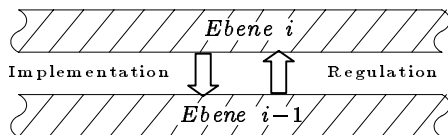


Bild 2.2: Zur Modell-Verfeinerung nach RAMMIG

Das so begründete „multi-level cybernetic model“ [15] erlaubt es, Kompositionsregeln für Entwurfsaktivitäten, wie Verkettung, Rekursion und Mehr-Ebenen-Rückkopplung, und klassische Entwurfsstrategien, wie *top-down*, *bottom-up* und Mischformen daraus, detaillierter zu modellieren.

2.2 Informationstheoretische Modelle

Gegenstand dieser Modellierungsklasse ist die Analyse des Informationsflusses, der während des Entwurfsprozesses auftritt. Die von SHANNON begründete *Mathematical Theory of Communication* [16] dient hierbei als formales Medium, um die zwei wesentlichen Aspekte eines Entwurfs — Komplexität und Kreativität — quantitativ zu untersuchen. Im ersten Ansatz wird die allen realen Systemen innewohnende *Kausalität* informationstheoretisch interpretiert. Roger CONANT [17], der diesen Ansatz 1976 entwickelt hat, faßt die Auslegung des Kausalitätsprinzips wie folgt zusammen:

„If each variable in a system is viewed as a message source which sends information about its values to the other variables which are 'listening', then what is conventionally seen as a network of causes and effects can be viewed as a network of transmitters, channels, and receivers. The strength of the causal relations can then be quantified by information rates T between variables or sets of variables. These rates are bounded by channel capacities. Information theory can therefore be used in the analysis of constraints or dependencies in multivariable systems, where it has particular advantages for dynamic systems and systems with hierarchical structure.“ (S. 254)

Obwohl in der Entwurfspraxis nicht von streng *stationären* (invariant gegenüber Zeitverschiebungen), geschweige denn von *ergodischen*⁶ Systemen ausgegangen werden kann und eine exakte Berechnung des Informationsflusses nur im Ausnahmefall möglich ist, besitzen die SHANNONSchen Lehrsätze ihren allgemeinen „Praxiswert“: Erkenntnisgewinn durch informelle Interpretation und Abschätzung der gefundenen Formeln und Identitäten. So gewann beispielsweise MILLER seine Erkenntnis von der Obergrenze der menschlichen Kanalkapazität durch informelle Anwendung des SHANNONSchen Begriffsrahmens und der darin aufgestellten Lehrsätze. Im Kontext technischer Entwurfssysteme ergeben sich praktische Richtlinien und Grenzwerte vor allem für die Synthese. Bevor wir aber genauer auf Informationsmodelle eingehen, stellen wir eine kurze Erläuterung der Grundbegriffe *Entropie*, *bedingte Entropie* und *Transmission* voran. Für ein tieferes Verständnis sei auf SHANNONS Abhandlung verwiesen.

⁶ WEAVER veranschaulicht den komplexen mathematischen Begriff der *Ergodizität* recht prägnant: „Es handelt sich um einen Vorgang, in dem eine Zeichenfolge erzeugt wird, die vielleicht der Traum eines Demoskopen wäre, weil jede einigermaßen große Probe danach strebt, repräsentativ für die Folge als Ganzes zu sein.“ [16, S. 21].

Der auf BOLTZMANN zurückgehende Begriff der *Entropie* erfaßt die *mittlere Information* einer diskreten Zufallsgröße x als Funktion der Wahrscheinlichkeiten p_i ihrer möglichen Ereignisse:⁷

$$H = -K \sum_{i=1}^n p_i \log p_i. \quad (2)$$

Übertragen steht Entropie für den Informationsgewinn, der in der Reduktion der Unbestimmtheit liegt (z. B. in der Kenntnis eines Versuchsausgangs). Wenn n für die Anzahl disjunkter Ereigniswerte steht, die x annehmen kann, so fällt $H(x)$ in das Intervall $[0, \log_2 n]$. Eine niedrige Entropie weist auf eine konzentrierte Wahrscheinlichkeitsverteilung, eine hohe auf eine diffuse. Sie ist Null, wenn ein Ereignis die Wahrscheinlichkeit Eins besitzt: sicheres Ereignis; sie ist maximal bei Gleichverteilung aller Ereignisse.

Werden mehrere Wahrscheinlichkeitsräume (x und y) betrachtet, so gilt bei gegenseitiger Abhängigkeit für die Entropie des kartesischen Produkts beider Räume (Notation nach SHANNON):

$$H(x, y) = H(x) + H_x(y) = H(y) + H_y(x). \quad (3)$$

Die Gleichung besagt, daß der Informationsgehalt aller Ereignis-Kombinationen der Zufallsvariablen x und y gleich dem Informationsgehalt einer der beiden ist plus der verbleibenden, d. h. *bedingten Entropie* der anderen, wenn erstere bekannt ist. (SHANNON bezeichnet $H_x(y)$ bzw. $H_y(x)$ auch als „equivocation“ und unterstreicht damit die durchschnittliche Mehrdeutigkeit des Ergebnisses.)

Als Maß für den Zusammenhalt (Abhängigkeit oder Koordinierung) zweier Zufallsvariablen wird die *Transmission* $T(x : y)$ über die Entropien definiert:

$$T(x : y) = H(x) + H(y) - H(x, y). \quad (4)$$

Sie wird Null, wenn x und y statistisch unabhängig sind, d. h. $H(x, y) = H(x) + H(y)$; sie wird maximal, wenn eine Variable die andere bestimmt, d. h. $H_y(x)$ bzw. $H_x(y) = 0$. Im Zusammenhang mit Übertragungskanälen erklärt sich die Transmission als die Information einer Nachrichtenquelle, vermindert um den Informationsverlust aufgrund einer gestörten Übertragung.

Informationsmodell nach CONANT

Roger CONANT [17] betrachtet ein System S als eine geordnete Menge von Zufallsvariablen: $S = \{x_1, x_2, \dots, x_n\}$. Von außen direkt beobachtbare Systemvariablen bilden die Menge S_o . S_{int} bezeichnet die verbleibenden internen Variablen, so daß $S = \{S_o, S_{int}\}$. E sei die Menge aller für das Systemverhalten relevanten Variablen der Umgebung. CONANT formuliert nun ein Teilungsge-

⁷ Aus historischen Gründen und wegen der weit verbreiteten Binärcodierung wird der Logarithmus dualis gewählt. Damit entfällt die positive Konstante K , und die Einheit der Entropie ist das *Bit*. Für kontinuierliche Variablen wäre eine Integraldefinition anzusetzen.

setz für die gesamte Informationsrate⁸ F in S :

$$F = F_t + F_b + F_c + F_n, \quad (5)$$

$$\begin{aligned} \text{mit: } F &= \sum_{j=1}^n \bar{H}(x_j) && \text{total rate,} \\ F_t &= \bar{T}(E; S_o) && \text{thrput rate,} \\ F_b &= \bar{T}_{s_o}(E; S_{int}) && \text{blockage rate,} \\ F_c &= \bar{T}(x_1 : x_2 : \dots : x_n) && \text{coordination rate,} \\ F_n &= \bar{H}_E(S) && \text{noise rate.} \end{aligned}$$

Die Komponenten von F weisen hierbei folgende Eigenschaften auf:

F repräsentiert die Gesamtaktivität im System.

F_t mißt den System-Durchsatz und ist ein Maß für die Stärke der Bindung zwischen Umgebung und Systemausgang. Als Kommunikationskanal gesehen wird F_t durch die Kanalkapazität C begrenzt.

F_b stellt die Rate dar, mit der Information aus der Umgebung innerhalb von S unterdrückt wird, um den Ausgang nicht zu beeinflussen. F_b hat als aktive „Filter-Rate“ große Bedeutung in informationsverarbeitenden Organismen. CONANT verweist hier auf Regelungsmechanismen in Natur und Technik, z. B. auf das psychologische Phänomen der *Gewöhnung* als herabgesetzte Reaktion auf wiederholte Stimuli, das hauptsächlich auf *Blockierung* zurückzuführen ist.

F_c steht für den erforderlichen Aufwand, um die Aktivität zwischen den Systemvariablen (Subsystemen) zu koordinieren. Sie kann als Maß für die interne Kommunikation in S angesehen werden, um ein Problem zu lösen. F_c ist das Charakteristikum eines Systems schlechthin: S agiert als Ganzes und nicht als Summe seiner Teile.

F_n schließlich repräsentiert den Rauschanteil am Gesamtfluß, die Informationsrate in S , wenn E vollständig bekannt ist. In der Kommunikationstechnik entspräche F_n dem thermischen Rauschen eines Kanals. Während CONANT diese Komponente zu Null setzt (deterministischer Fall), sieht Cees KOOMEN hier den kreativen Anteil des Entwerfers am Entwurfsprozeß. Wir werden darauf noch zurückkommen.

Aus der additiven Eigenschaft des CONANTschen Teilungsgesetzes (5) ergeben sich bei vorgegebenem F (*man power*, Rechnerkapazität) konkurrierende Freiheitsgrade für die Komponenten F_t , F_b und F_c . Verfolgt man das *ökonomische Prinzip* — Maximierung des Ertrages bei Minimierung des Aufwands —, so lassen sich aus der Analyse von F Richtlinien für die Systemkonzeption ableiten:

..1) produce the minimum allowable output ('do not do anything unnecessary'). 2) perform as little blockage as possible ('try to take in a minimum of irrelevant input'). 3) reduce internal coordination to the minimum consistent with other requirements ('maximize freedom of the components'). 4) as far as possible, match components to tasks so that each component is operated at capacity ('let each component do what it does best, and work it as hard as you can')." (S. 248)

Diese Konstruktionsvorschriften liefern zugleich

⁸ Um die Historie der Ereignisse einer Zufallsvariablen x zu erfassen, führt CONANT *Entropie-Raten* ein (angedeutet durch Überstriche):entropy of x conditional on all its prior values, which is the information carried per observation in a long sequence. The total uncertainty of a long sequence $\langle x(t), x(t+1), \dots, x(t+n) \rangle$ is then (approximately) this entropy rate times the sequence length" (S. 244).

informationstheoretische Argumente für SIMONS hierarchisches Strukturierungs-Postulat:

$$\bar{H}(S) = \sum_{i=1}^N \bar{H}(S^i) - \bar{T}(S^1 : S^2 : \dots : S^N). \quad (6)$$

In Worten: Die Entropie(rate) eines Systems S ist im allgemeinen geringer als die Summe der Entropien seiner Subsysteme S^i . Während SIMON seine These, daß die darwinistische Evolution Organismen mit isolierbaren Strukturen begünstige, auf Stabilitäts-Überlegungen stützt, erlaubt die Informationstheorie die gleiche Schlußfolgerung über die statistische Untersuchung des Informationsflusses: Höchste Ökonomie (ein Selektionskriterium der Evolution) bedingt eine möglichst geringe Transmissionsrate zwischen den Subsystemen. Dies entspricht SIMONS Theorem der 'niederfrequenten Dynamik' auf den Inter-Komponentenverbindungen.

Informationsmodell nach KOOMEN

Cees KOOMEN [18, 19] analysiert den Entwurfsprozeß als geordnete Transformationsfolge von Systemmodellen $\langle M_p, M_{p+1}, \dots, M_q \rangle$, wobei ein Modell M_i das Gesamtsystem auf dem Beschreibungsniveau i repräsentiert. Als „modeling primitive“ definiert er das *Detailing-step*-Paradigma: Der Entwurfsprozeß verfolgt die sukzessive Detaillierung der Beschreibungsmodelle bis zum Grad des Zielmodells M_q . Ein Verfeinerungs-Schritt beinhaltet die Entwurfsaktivität, um von einer Spezifikation S der geforderten System-Eigenschaften (Ausgangsmodell M_i) zu einer Implementierung I (Entwurf von M_{i+1}) zu gelangen. Bild 2.3 verdeutlicht diese Betrachtungsweise:

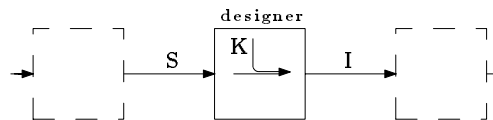


Bild 2.3: SKI-Modell eines Entwurfsabschnitts nach KOOMEN

Mit Einführung der Komponente K — des Wissens, das vom Entwerfer geschaffen oder als Ressource eingesetzt wird, um I von S abzuleiten — verallgemeinert KOOMEN das Informationsmodell von CONANT auf nicht-deterministische Systeme. Die Rauschrate F_n wird substituiert durch den Einsatz und die Kreativität des Entwerfers: $F_n \leftarrow \bar{H}_S(I)$. „Hence, the designer's "noise rate" reflects the unexpected, the unpredictable, the creative act.“ [19, S. 21].

Auf der Grundlage des *Detailing-step*-Paradigmas können informationstheoretische Aussagen über Eigenschaften des Entwurfsprozesses gewonnen werden. Das Ausgangsrauschen einer Modell-Transformation $\bar{H}_S(I)$ ist größer als Null, wenn eine nicht-triviale Implementation vorliegt. Da die Energie für den Implementierungs-Prozeß natürlich endlich ist, gilt insgesamt $0 < \bar{H}_S(I) < \infty$. Das Wissen K , um von einer Spezifikation S zu einer Implementation I zu gelangen, muß letztlich der Bedingung

$$\bar{H}_{S,K}(I) = 0 \quad (7)$$

genügen. Wegen der Ähnlichkeit zum Energie-Erhaltungssatz der Thermodynamik bezeichnet KOOMEN diesen Ausdruck auch als „first information law for system design“ (S. 25). Die Gleichung impliziert, daß K durch eine Menge von Symbolen ausgedrückt werden kann, die dem Beobachter bekannt sind. Falls K in einer formalen Sprache beschreibbar ist (vergleiche mit AMKREUTZ' Begriff der Quantifizierbarkeit), so kann die Modell-Transformation mit deterministischen Maschinen erfolgen.

Interpretiert man den Entwerfer als ein informationsverarbeitendes System (d. h. unter anderem, daß seine CONANTSche Informationsrate F begrenzt, seine kreativen und intellektuellen Fähigkeiten endlich sind), so läßt sich folgende Einschränkung formulieren:

$$\dot{H}_S(I) \leq C_d(S, \tau_d). \quad (8)$$

Hierbei steht $C_d(S, t)$ für die Informationsmenge, die der Entwerfer in einer Zeit t erzeugen kann, um eine Implementation von S zu finden. Bei vorgegebener Zeit τ_d für die Problemlösung setzt (8) die erzeugte Information in Relation zur Kapazitätsfunktion C_d . Da durch die Relation eine obere Grenze für die Entropie-Änderung gegeben ist, bezeichnet KOOMEN dies auch als „second information law for system design“ (S. 25) in Anlehnung an den Entropie-Richtungssatz der Thermodynamik. Für den Fall, daß $\dot{H}_S(I) > C_d(S, \tau_d)$, d. h. der Entwerfer ist nicht in der Lage, S zu implementieren, ergeben sich drei Lösungsmöglichkeiten:

- Dem Entwerfer wird zur Lösungsfindung mehr Zeit gegeben (τ_d erhöhen).
- Die Entwurfskomplexität wird erniedrigt, indem hierarchische Zwischenebenen eingeführt werden, so daß Implementierungen auf diesen Ebenen der obigen Bedingung genügen.
- $C_d(S, \tau_d)$ wird erhöht, entweder durch mehr Personal oder durch Rechnereinsatz.

KOOMEN untersucht des weiteren die informationsverarbeitenden Aspekte beim Einsatz von Maschinen und analysiert den Informationsfluß durch die Mensch-Maschine-Schnittstelle. Bezüglich Einzelheiten sei auf [19, S. 26f] verwiesen.

Zusammenfassend erlauben die informationstheoretischen Erkenntnisse von KOOMEN und CONANT die Formalisierung der Grundmechanismen im Entwurfsprozeß. Sie helfen, Begriffe, wie Kreativität, Komplexität und Modellinformation, zu klären. Darüber hinaus präzisieren sie elementare Konstruktionsvorschriften für einen *ökonomischen* Systementwurf.

2.3 Verfahrensmodelle

In die dritte Kategorie für Entwurfsmodelle ordnen wir diejenigen ein, die den Entwurfsprozeß unter einem bestimmten Blickwinkel — *sichtenorientiert*⁹ — betrachten. Da dies vor dem Hintergrund vielfältiger Verfahrenslandschaften ge-

⁹ Als ein Beispiel sichtenorientierter Modellierung im VLSI-Entwurf sei auf das Datenaustausch-Format EDIF verwiesen (siehe z. B. [20]), das die Entwurfsdaten unter verschiedenen Sichten beschreibt: *viewTypes* wie *schematic*, *netlist* und *maskLayout*.

schieht (Methoden, Techniken und Werkzeuge), sprechen wir hier von praxisorientierten oder verfahrenstechnischen Modellen. Die Sichtenorientierung erleichtert den Zugang zu hochkomplexen Entwurfsprozessen und wirkt einer Einengung der Modellierungsgenauigkeit entgegen. Repräsentativ befassen wir uns mit dem weit verbreiteten *Y-Diagramm*, das von Daniel GAJSKI und Robert KUHN [21] 1983 entwickelt, von Robert WALKER und Donald THOMAS [22] 1985 verfeinert und 1988 von GAJSKI [9] zuletzt aktualisiert wurde.

Y-Diagramm nach GAJSKI et al.

Der Anspruch des Y-Diagramms — „a comprehensive view of VLSI CAD tools“ [21, S. 12] — wurzelt in dem Bedürfnis, die Hauptströmungen in der Chip-Konstruktion der 80er Jahre („school of thoughts“) übersichtlich zu erfassen. Je nach dem Grad der Automatisierung und der Art der Wissenseinbringung unterscheiden GAJSKI und KUHN drei Verfahrenslandschaften:

Entwurfsunterstützung: Alle Entwurfsentscheidungen obliegen dem Entwerfer. CAD-Werkzeuge wirken ausschließlich unterstützend: „efficient paper and pencil“.

Expertensysteme: Verlagerung der Entwurfsunterstützung von den mechanischen zu den intelligenten Aspekten des Entwurfs — Expertenwissen als Quelle wird in Form von Entwurfsregeln in Wissensbanken abrufbar gespeichert (effizient für die Entwurfsprüfung, noch in den Anfängen bezüglich Entwurfssynthese und -planung).

Entwurfsautomation: In der Zielsetzung konträr zum CAD-Ansatz: statt zu unterstützen, wird der teilweise oder gänzliche Verzicht auf den Entwerfer angestrebt (z. B. automatisches Platzieren und Entflechten von Layout-Zellen, Silicon Compiler).

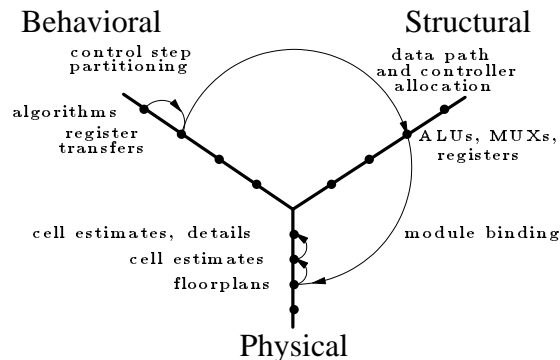


Bild 2.4: Y-Diagramm mit Anwendungsbeispiel nach WALKER und THOMAS

Das Y-Diagramm soll also eine Antwort geben auf die Frage: „Wie werden VLSI-Werkzeuge eingesetzt, um eine Entwurfsspezifikation auf höherer Ebene zu verifizieren und in eine Implementation auf niedriger Ebene zu überführen?“ Einerseits sollen die Verfahrensketten (Synthese, Analyse und Verifikation) bereits in der Entwurfsplanung veranschaulicht werden, und zwar im gesamten Entwurfsraum sichtenorientiert und unter Berücksichtigung der geltenden Entwurfsab-

straktion. Andererseits soll das Y-Modell die Mechanismen und Interaktionen *realer* Entwurfssysteme transparent machen. Bild 2.4 illustriert letzteres für das von WALKER und THOMAS angegebene Beispiel der automatischen Datenpfad-Synthese [22, S. 248].

Der Entwurfsprozeß wird schematisch als Axial-Modell dargestellt: Die Achsen spannen den Entwurfsraum in drei komplementäre Entwurfsbereiche auf, als Sichten oder Domänen bezeichnet. Die *Verhaltenssicht* ist auf die Funktionalität, die *Struktursicht* auf die logische Struktur und die *physikalische Sicht* auf die physikalische Implementierung gerichtet. Andere Aspekte¹⁰ können unter diesen Domänen subsumiert werden. Entlang der Raumachsen erfolgt von außen nach innen die Entwurfsverfeinerung auf Ebenen zunehmender Detaillierung. Mittels einer graphischen Notation lassen sich die Entwurfswerkzeuge nach Art und Wirkungsbereich im Entwurfsraum lokalisieren. Synthetisierende Werkzeuge werden als gerichtete Übergänge notiert: aus der Verhaltens- in die Struktur-Domäne (im Bild 2.4 „data path and controller allocation“), aus der Struktur- in die physikalische Domäne („module binding“) oder von einer höheren Abstraktionsebene in eine niedrigere gleicher oder benachbarter Domäne („control step partitioning“). Umgekehrte Übergänge beziehen sich auf entsprechende analysierende Werkzeuge (z. B. Schaltkreis-Extraktoren). Eine Schleife (Anfangs- und Endpunkt fallen zusammen) repräsentiert ein Analysewerkzeug, das ausschließlich auf einer Ebene arbeitet (Simulatoren). Darüber hinaus erlauben Einstiegs- und Zielpunkt der Verfahrenskette das Identifizieren der ursprünglichen Entwurfsspezifikation (hier die algorithmische Verhaltensbeschreibung) und der erreichten Implementierungstiefe (Masken-Layout).

Ein weiteres Charakteristikum des Y-Modells ist die 'non-isomorphe' hierarchische Dekomposition der Entwurfskomponenten: 1:n-Abbildungen zwischen Verhaltens-, Struktur- und physikalischen Entwurfsobjekten. Im Beispiel von Bild 2.4: Für eine gegebene Menge von Register-Transfer-Beschreibungen existieren mehrere abstrakte Implementierungen in der Struktur-Domäne. „Choosing one of these alternatives is exploring the design space.“ (S. 458) Weiterhin erlaubt das Modell die Lokalisierung von Entwurfseinschränkungen, wie beispielsweise Flächenbedarf und Verlustleistung.

Als „inter domain links“ verlaufen die Repräsentationsebenen konzentrisch um den Ursprung des Y-Diagramms. GAJSKI spezifiziert einen Standardsatz von fünf Abstraktionsebenen¹¹ und nennt exemplarisch einige Entwurfsobjekte der jeweiligen Sicht (Tabelle 2.5).

¹⁰ RAMMIG regt zum Beispiel die Erweiterung des Y- zum X-Diagramm an, um testspezifische Aspekte im Entwurfsraum aufzuzeigen [14, S. 14f].

¹¹ Konrad WEBER schlägt für die Schichtung der Abstraktionsebenen den Begriff der „VLSI-Stratifikation“ vor und spricht in diesem Zusammenhang von *stratifizierten* Entwürfen. [23, S. 14f]

	Domains of Description		
	behavioural	structural	physical
1	performance specs.	CPUs, memories, controllers, busses	physical partitions
2	algorithms	hardware modules, data structures	clusters
3	register transfers, state sequencing	ALUs, MUXs, registers, microsequencer	floorplans
4	Boolean equations	gates, flip-flops, cells	cell, module plans
5	transfer functions, timing	transistors, wires, contacts	layout

Tab. 2.5: VLSI-Stratifikation versus Entwurfssichten nach GAJSKI [9, S. 5] (1: system, 2: algorithmic, 3: micro-architectural, 4: logic, 5: circuit level)

2.4 Einordnung der Vorschläge

Herbert STACHOWIAK kommt das Verdienst zu, als erster den Modellbegriff wissenschaftlich fundiert zu haben. Seine *Allgemeine Modelltheorie* [24] erlaubt die Charakterisierung und Klassifizierung der unüberschaubaren Vielfalt wissenschaftlicher Modelle. Um eine möglichst prägnante Gegenüberstellung der diskutierten Modellvorschläge zu erreichen, werden wir im folgenden die Erkenntnisse STACHOWIAKS schematisch anwenden. Zuvor aber sei die Allgemeine Modelltheorie kurz umrissen:

„Etwas schärfer gefaßt: Modelle sind zwar immer Modelle von etwas. Abbildungen, Repräsentationen natürlicher oder künstlicher Originale (die selbst wieder Modelle sein können). Aber sie erfassen im allgemeinen *nicht alle* Originalattribute, sondern stets nur solche, die für die Modellbildner und/oder Modellverwender relevant sind. Modelle sind mithin ihren Originalen nicht per se zugeordnet: sie erfüllen ihre Ersetzungsfunktion stets a) für bestimmte Erkenntnis- und/oder Aktionssubjekte, b) innerhalb bestimmter Zeitintervalle der Originalrepräsentation und c) relativ zu bestimmten Zwecken und Zielen, denen die Modellbildung und die Modelloperationen unterliegen. Modellbildungen unterliegen hiernach dem Frage-schema: Modell *wovon*, *für wen*, *wann* und *wozu*.“ [25, S. 118]

Damit sind die drei Hauptmerkmale des allgemeinen Modellbegriffs genannt: *Abbildungs-*, *Verkürzungs-* und *pragmatisches Merkmal*. Das VENN-Diagramm in Bild 2.6 verdeutlicht die Zusammenhänge:

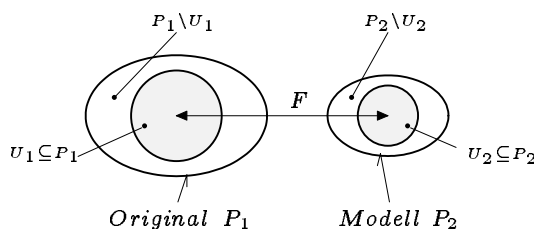


Bild 2.6: Original-Modell-Abbildung nach STACHOWIAK

Die Abbildungsfunktion F ordnet Modell-Attribute Original-Attributen zu. (Die Modelltheorie nach STACHOWIAK faßt *Original* und *Modell* als Prädikatenklassen P_1 und P_2 auf und stellt sie in eine mengentheoretische Relation zu einander.) Für unsere Anwendung genügt es, unter dem Abbildungsmerkmal die Domäne — Umfang und Funktionalität — des modellierten Realitätsausschnitts zu verstehen.

Das Verkürzungsmerkmal bezieht sich auf die Menge der *präterierten*, d. h. auf die von F nicht erfaßten Attribute $P_1 \setminus U_1$, und der *abundanten*, d. h. im Original nicht vorkommenden Attribute $P_2 \setminus U_2$, wobei U_1 den *Abbildungsvorbe-*

reich und U_2 den *-nachbereich* bezeichnen. Die Modell-Verkürzung ist also ein (negatives) Maß für die Realitätstreue, für die Ähnlichkeit zwischen Original und Modell. Sie gibt Aufschluß über die Bijektivität der Klassen P_1 und P_2 . Für unsere Zwecke verbinden wir mit diesem Begriff die Verkürzung auf bestimmte Aspekte des Originals: system-, entwerfer- oder benutzerorientierte Perspektiven. Weiterhin unterscheiden wir hier die Art der Modellaussagen, ob qualitativ, quantitativ oder phänomenologisch.

Das pragmatische Merkmal schließlich berücksichtigt: a) die Zielsetzung (Beschreibungs-, Erklärungs- oder Entscheidungsmodell), b) das Zeitintervall (Gültigkeit, technologische Annahmen) und c) die Leistungen des Modells (z. B. im Rahmen einer Entwurfsmethodik).

Gegenüberstellung

Als Abschluß unserer Rückschau stellen wir mit Hilfe obiger Charakterisierung die diskutierten Modellierungsvorschläge einander gegenüber. Neben den drei Hauptmerkmalen Abbildung *A*, Verkürzung *V* und Pragmatik *P* kommt ein weiteres hinzu: das Konstruktionsmerkmal *K*, welches zur Klassifikation der Modelle dient. Kybernetische, informationstheoretische und verfahrenstechnische Modelle unterscheiden sich in ihrer Notation und ihrem Gegenstandsbereich. Tabelle 2.7 faßt die Charakteristika der Modelle vergleichend zusammen:

Modelle des VLSI-Entwurfsprozesses			
K	Kybernetik	Informationstheorie	Verfahrenstechnik
A	kybernetische Prozeßelemente: evaluation, decision, regulation (feedback)	Informationsfluß, Entropie, Transmission, Kanalkapazität	CAD/CAE-Instrumentarium, Verfahrensketten und -landschaften
V	systemorientiert qualitativ: bewerten, entscheiden, steuern	system- und entwerferorientiert quantitativ: Formeln, Identitäten, Eckwerte	benutzer- und praxisorientiert phänomenologisch: VLSI-Objekte, Tools
P	Entscheidungsmodell allgemeingültig Strukturieren der Entwurfsaufgabe, Identifizieren der Entwurfsaktivitäten und deren Wirkungsgefüge	Erklärungsmodell allgemeingültig Konstruktionsprinzipien, Analyse von Kreativität und Komplexität im Entwurfsprozeß, Grenzwertbetrachtungen	Beschreibungsmodell technologieabhängig Identifizieren und Lokalisieren der Entwurfswerkzeuge und -handlungen, Orientierungshilfe

Tab. 2.7: Gegenüberstellung der diskutierten Modellvorschläge

Als Quintessenz unserer Studie können wir festhalten: Systemtheoretische und kognitive, d. h. die begriffliche Vorstellungsweise des Modellbenutzers unterstützende Aspekte werden nicht von einem Modell allein reflektiert. Was fehlt ist eine *monistische Weltsicht* — ein Modellierungskonzept, das die Anforderungen einer durchgängigen Entwurfsmethodik, eines abstrahierten Systemdenkens und globaler Datenintegrität einheitlich unterstützt. Im Ausblick werden wir unsere Vorstellungen präzisieren.

In geraffter Form soll eine objektorientierte Entwurfsmethodik vorgestellt werden, wobei wir uns haben leiten lassen von den eingangs beschriebenen Prinzipien der Komplexitätsbewältigung und der Synergie obiger Modellierungsvorschläge. Unser Ansatz wurzelt im Motivationsumfeld des Forschungsprojekts DASSY¹² und der EDIF TSC Information Modelling Group¹³, in dem die Autoren aktiv involviert sind. Das Problemfeld «VLSI-Entwurf» ist nach unserer Erfahrung gekennzeichnet durch fortgesetzte Fehlanpassungen zwischen Problembewußtsein, -formulierung und -lösung. Der Entwurfspfad von der Idee zum Produkt führt über mehrere kognitive und semantische Einbrüche: vom Problembewußtsein als kognitivem Prozeß über die Problemformulierung in einer formalen Notation (in Sprachen und Modellen) zur Problemlösung durch Abbilden auf Objekte und Prozesse des Implementierungsmediums (der Programmierung und Datenhaltung). Je größer die Sprünge, desto schwieriger das Problem, Datenkonsistenz global zu gewährleisten. Aus dieser Einsicht entsteht das Bedürfnis nach einer übergreifenden durchgängigen Entwurfsmethodik. Wir sind der Überzeugung, daß durch Einführen des *Objekt-Paradigmas* in den Entwurfsprozeß der duale Aspekt der VLSI-Konsistenzsicherung — methodisch und operativ — berücksichtigt werden kann.

3.1 Motivation eines Objekt-Modells

Die Parallelen zwischen Hardware- und Software-Entwurf sind offensichtlich: Ob *system on chip* oder *programming in the large*, in beiden Fällen handelt es sich um das Unterfangen, hochkomplexe Systeme überprüfbar zu entwerfen. Hardware-Aspekte, wie Flächenbedarf, Laufzeit, Gatter-Äquivalente, Test-Strukturen, werden in der Regel erst auf implementierungsnahen Ebenen einbezogen; sie stellen physikalische Parameter bzw. Entwurfsregeln für den Software-Entwurf dar. Eingedenk dieser Analogie führen wir die Objektorientierung als probates Mittel zur Komplexitätsbewältigung in den VLSI/ULSI-Entwurfsprozeß ein. Wir begründen dies mit dem breiten theoretischen Fundament, auf dem die objektorientierten Konzepte¹⁴ ruhen.

¹² DASSY, ein vom Bundesministerium für Forschung und Technologie gefördertes Projekt in Industrie-Kooperation, verfolgt das Ziel, Schnittstellen in den Bereichen Entwurf, Test und Fertigung integrierter Schaltungen zu standardisieren. Die Standardisierung bezieht sich zum einen auf Datentransfer-Formate; zum andern soll eine Standard-Werkzeugschnittstelle zwischen CAx-Entwurfswerkzeugen und einer gemeinsamen Datenbasis realisiert werden.

¹³ Wegen der unüberschaubaren Fülle und Vielfalt der Entwurfsdaten hat sich das EDIF Technical Sub-Committee IMG zum Ziel gesetzt, den Informationsgehalt und die Informationsstrukturen der durch EDIF beschriebenen Entwurfsdaten einheitlich zu modellieren (für den Bereich *Test* siehe z. B. [26]).

¹⁴ Der interdisziplinäre Charakter nährt wohl auch das „populäre (Un)wissen“ über die Bedeutung der Objektorientierung. Was die Programmiersprachen betrifft, so prophezeite T. RENTSCH bereits 1982: „My guess is that object-oriented programming will be in the 1980s what structured programming was in the 1970s. Everyone will be in favor of it. Every manufacturer will promote his products as supporting it. Every manager will pay lip service to it. Every programmer will practice it (differently). And no one will know just what it is.“ [27, S. 51]

Philosophie: Das Objekt-Konzept ließe sich bis in die griechische Philosophie zurückverfolgen, z. B. auf DEMOKRITS atomistische Weltansicht, in der unteilbare *Dinge* im Mittelpunkt des Interesses stehen. Einen aktuellen philosophischen Bezug gibt YAIR WAND: Er beantwortet schlüssig die Frage

„Is the emergence of the object paradigm an *empirical indication* that humans find it easier to describe perceptions of the world through the notion of objects?“ [28, S. 538].

indem er ein Objekt-Modell formal aus der Ontologie ableitet (der Wissenschaftsphilosophie von den Ordnungs-, Begriffs- und Wesensbestimmungen des Seienden).

Psychologie: Hier sei zum einen auf die von MILLER inspirierte Kognitionswissenschaft [3, 4] verwiesen (Unterstützen kognitiver Prozesse durch Klassifikation des Gegenstandsbereichs). Zum anderen schlägt Marvin MINSKY in *The Society of Mind* [29] ein objektorientiertes Modell der menschlichen Intelligenz vor: Verstand konstituiert sich aus einer „society of mindless agents“. Erst durch deren mechanisches Wechselspiel finden wir den Zugang zum Phänomen der Intelligenz. Weiterhin sei auf das *Objekt-Konzept* in der psychologischen Entwicklung eines Kindes hingewiesen: Jean PIAGET [30] beobachtet, daß ein Kind mit dem ersten Lebensjahr die Vorstellung von *Objekt-Permanenz* entwickelt und Objekte zunehmend hierarchisch klassifiziert.

Systemtheorie: Hier sei das hierarchische Strukturierungsprinzip von SIMON [5, 6] genannt, das sich in den Vererbungs- und Instanzen-Hierarchien eines Objekt-Modells widerspiegelt. Auch drängt sich der Vergleich mit dem Begriff des MEALEY-Automaten auf. Die begriffliche Analogie zwischen Objekten und endlichen Automaten läßt sich aus der Objektdefinition ableiten: Zustand, Verhalten und Identität. Informationstheoretische und kybernetische Parallelen werden noch deutlich, wenn wir das Objekt-Paradigma im ganzen darstellen.

Informatik: Die bewährten Konzepte des Software Engineering, wie strukturierter Entwurf, Modularisierung, abstrakte Datentypen und Nebenläufigkeit, finden ihr objektorientiertes Pendant. Mehr noch: Objektorientierung ebnet einen durchgängigen *pragmatischen* Entwurfspfad von der Konzeptionsphase bis zur Implementierung. Die Phasen Analyse, Entwurf, Programmierung und Datenhaltung genügen jede für sich dem Objekt-Paradigma. Die konzeptionellen Objekte der *problem domain* sind auch die Software-Objekte der *implementation domain*. Zwischen konzeptionellen Schemata und implementierungsnahen Datenmodellen sind 1:1-Abbildungen möglich.

Das Objekt-Paradigma

Der noch verbleibende Raum verlangt eine gedrängte Beschreibung: Wir sehen als implementierungsunabhängige Merkmale der Objektorientierung die Abstraktionen *Kapselung* und *Klassifikation* und die Mechanismen zur Ressourcenteilung *Exemplarbildung* und *Vererbung*. Sie fus-

sen auf der monistischen¹⁵ Auffassung des Objektbegriffs. Ein Objekt verfügt autonom über Struktur und Verhalten. Es ist konzipiert nach den Prinzipien des *information hidings* und des *abstrakten Datentyps*, d. h. die Realisierung der Objekteigenschaften bleibt der Umwelt verborgen. Der Objektzustand ist nur über interne Operationen veränderbar. Zustandsänderungen können allein durch *Nachrichtenaustausch* zwischen den Objekten erfolgen.

Der Begriff der Kapselung steht in der programmiertechnischen Tradition der Wiederverwendung funktionaler Software-Moduln (Prozeduren, Makros, Bibliotheken). Objektorientierte Techniken verallgemeinern den Container-Begriff der Kapselung, erhöhen das *code sharing*, indem nicht nur Programme sondern auch die beeinflussbaren Daten gekapselt werden.

Gemäß ihrer durch Abstraktion gewonnenen gemeinsamen Merkmale werden Objekte in Klassen zusammengefaßt. Klassifizierte Objekte verfügen über verwandte Operationen und zeichnen sich somit durch uniforme Verhaltensmuster aus. Für den Teilungs-Mechanismus der Exemplarbildung fungiert die Klasse als Schablone, mit deren Hilfe neue Objekte, sogenannte *Instanzen*, erzeugt werden können. Eine Schablone beinhaltet alle für die Klassenzugehörigkeit relevanten Deklarationen passiver und aktiver Eigenschaften. Einheitlichkeit und Unabhängigkeit der schablonisierten Instanzen werden garantiert.

Vererbung als ein Mittel der Software-Strukturierung und -wiederverwendung ermöglicht die Definition neuer Klassen, die die Merkmale der übergeordneten Klassen erben. Vererbung fördert somit die Bildung von Klassen-Hierarchien. Spezielle Vererbungsaspekte erlauben die Modellierung komplizierter Abhängigkeits-Strukturen: einfache oder multiple Vererbung, falls eine Unterklasse von einer oder mehreren Oberklassen erbt; selektive Vererbung, falls abgeleitete Eigenschaften überschrieben oder ausgeblendet werden sollen. Eine dezidierte Vererbungsstrategie sichert darüber hinaus die Datenkonsistenz im Ableitungsgraphen. Hier stellen wir uns vor, daß durch Vererbung von Initialwerten, Entwurfsregeln, Constraints usw. die Gesamtkonsistenz eines Datenbestandes dezentral verbessert werden kann.

3.2 Ansatz einer objektorientierten Entwurfsmethodik

Objektorientierung als Leitmotiv einer VLSI-Entwurfsmethodik führt auf die von Grady BOOCH postulierte „kanonische Form“ wohlstrukturierter komplexer Systeme: Klassen-Strukturen als *kind-of-* und Objekt-Strukturen als *part-of-*Hierarchien [32, S.12f]. Die grundsätzliche Vorgehensweise bezeichnet BOOCH als *round-trip gestalt design*:

„A style of design that emphasizes the incremental and iterative development of a system, through the refinement of different yet consi-

¹⁵ Die monistische Weltansicht geht ausschließlich von Populationen kooperierender Objekte aus. Im Gegensatz dazu stünde zum Beispiel die duale Weltansicht des *strukturierten Entwurfs*: Daten- und Steuerflüsse.

stent logical and physical views of the system as a whole; the process of object-oriented design is guided by the concepts of round-trip gestalt design; round-trip gestalt design is a recognition of that fact that the big picture of a design affects its details, and that the details often affect the big picture." (S. 517)

Das zentrale Problem liegt in der Dekomposition in geeignete Objekte bzw. deren Klassen. BOOCH stellt die objektorientierte Dekomposition der algorithmischen gegenüber:

„algorithmic decomposition: The process of breaking a system into parts, each of which represents some small step in a larger process. The application of structured design methods leads to an algorithmic decomposition, whose focus is upon the flow of control within a system.“ (S. 512)

„object-oriented decomposition: The process of breaking a system into parts, each of which represents some class or object from the problem domain. The application of object-oriented design methods leads to an object-oriented decomposition, in which we view the world as a collection of objects that cooperate with one another to achieve some desired functionality.“ (S. 516)

Die eingefahrene algorithmische Partition des Steuerungsfusses hat die klassischen Entwurfswerkzeuge der Synthese und Verifikation hervorgebracht. Konzeptionell lassen sie sich dennoch, wie James DANIELL in seiner Abhandlung *An Object Oriented Approach to CAD Tool Control* [33] zeigt, als abstrakte Objekte im Sinne der Objektorientierung auffassen.

Der objektorientierte Entwurf umfaßt neben dem Prozeß der Dekomposition auch eine entsprechende Notation (*boochgrams*), die es erlaubt, die logisch-physikalischen sowie die statisch-dynamischen Modelle des zu entwerfenden Systems zu beschreiben. Als Ergebnis dieser Entwurfsmethodik ergeben sich Klassen, Beziehungstypen und Vererbungsstrategien, die bei einer objektorientierten Implementierung „1:1“ auf Programm- und Datenbank-Konstrukte abbildbar sind.

Weitere Arbeiten

Zunächst gilt es, geeignete VLSI-Basisklassen zu identifizieren. Diese müßten vor allem die Entwurfsdomänen Verhalten, Konnektivität und Graphik abdecken. Als gemeinsames Strukturmuster für Klassen und Objekte betrachten wir die »Zelle«. Sie stellt den Grundbaustein der Entwurfsmodellierung und der Implementierung dar — parallel zur Zell-Doktrin der Biologie, nach SIMON „a law of qualitative structure“ [31, S. 290ff.]. Charakteristisch für eine Zelle ist ihre Dichotomie: öffentliches Nachrichtenprotokoll (Schnittstelle) und private Implementierung. Entwurfssichten werden durch Spezialisierung repräsentiert: Unterklassen. Abstraktionsebenen spiegeln sich als Generalisierungsstufen in multiplen Vererbungslinien wider: Oberklassen. Derzeitige Arbeiten beschäftigen sich mit der Identifikation ebenenspezifischer Objekte. Klassen- und Objektdiagramme sollen graphisch in der BOOCH-Notation entwickelt werden.

Zum ändern geht es bei den weiteren Arbeiten um den zentralen Begriff der »Konsistenz«. Es gilt, möglichst alle praxisrelevanten Facetten der Konsistenz im Entwurfsprozeß zu definieren. Darauf aufbauend sollen objektorientierte Vererbungsstrategien zur Konsistenzsicherung traditionellen Konzepten gegenüber gestellt werden, wie *constraint propagation*, *event/trigger-Mechanismus* u.a.m.

Benutzte Literatur

- [1] MEAD, Carver; CONWAY, Lynn: *Introduction to VLSI Systems*. Addison-Wesley Publishing Company, Reading, Mass., 1980
- [2] WEAVER, Warren: *Science and Complexity*. American Scientist, Jg. 36, 1948, S. 536–544
- [3] MILLER, George A.: *The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information*. In: *The Psychological Review*, Jg. 63, H. 2, 1956, S. 81–97
- [4] GARDNER, Howard: *The Minds' New Science: A History of the Cognitive Revolution*. Basic Books Inc., New York, 1985
- [5] SIMON, Herbert A.: *The Sciences of the Artificial*. MIT Press, Cambridge, Mass., 1982
- [6] SIMON, Herbert A.: *The Architecture of Complexity*. Proceedings of The American Philosophical Society, Jg. 106, H. 6, 1962, Nachdruck in: *General Systems*, Jg. 10, 1965, S. 63–76
- [7] ROPOHL, Günter: *Modelle im Technikunterricht*. In: *Modelle und Modelldenken im Unterricht: Anwendung der Allgemeinen Modelltheorie auf die Unterrichtspraxis*, Hrsg: Stachowiak, Herbert, Verlag Julius Klinkhardt, Bad Heilbrunn/Obb., 1980, S. 123–143
- [8] SOMMERVILLE, Ian: *Software Engineering*. Addison-Wesley Publishing Company, Reading, Mass., 1990
- [9] GAJSKI, Daniel D.: *Silicon Compilation*. Addison-Wesley Publishing Company, Reading, Mass., 1988
- [10] MÜLLER, Roland: *Zur Geschichte des Modelldenken und des Modellbegriffs*. In: *Modelle — Konstruktion der Wirklichkeit*, Hrsg: Stachowiak, Herbert, Wilhelm Fink Verlag, München, 1983, S. 17–86
- [11] SACHSSE, Hans: *Einführung in die Kybernetik*. F. Vieweg, Braunschweig, 1971
- [12] WIENER, Norbert: *Cybernetics: Or Control and Communication in the Animal and the Machine*. MIT Press, Cambridge, Mass., 1948
- [13] AMKREUTZ, J. H.: *Cybernetic Model of the Design Process*. In: *Computer Aided Design*, Jg. 8, H. 3, 1976, S. 187–192
- [14] RAMMIG, Franz J.: *Systematischer Entwurf digitaler Systeme*. B. G. Teubner, Stuttgart, 1989
- [15] RAMMIG, Franz J.: *A Multilevel Cybernetic Model of the Design Process*. In: *Methodologies for Computer System Design*, Hrsg.: Giloi, W. K.; Shriver, B. D., Elsevier Science Publishers B. V. (North-Holland), IFIP, 1985, S. 87–103
- [16] SHANNON, Claude E.; WEAVER, Warren: *Mathematical Theory of Communication*. University of Illinois Press, Urbana, 1949; Deutsche Übersetzung von Dreßler, Helmut: *Mathematische Grundlagen der Informationstheorie*. Oldenbourg Verlag, München, Wien, 1976
- [17] CONANT, R. C.: *Laws of Information Which Govern Systems*. In: *IEEE Trans. on Systems, Man, and Cybernetics*, Jg. 6, H. 4, 1976, S. 240–255
- [18] KOOMEN, Cees J.: *Information Laws for System Design*. In: *Proc. Intern. Conf. on Cybernetics and Society*, Tokyo, Bd. II, 1978, S. 1013–1017
- [19] KOOMEN, Cees J.: *The Entropy of Design: A Study on the Meaning of Creativity*. In: *IEEE Trans. on Systems, Man, and Cybernetics*, Jg. 15, H. 1, 1985, S. 16–30
- [20] QUBELDEY-CIRKEL, Klaus; WAHL, Michael G.: *The Electronic Design Interchange Format EDIF: Capabilities and Evaluation*. In: *Proc. of the Int. Conf. on the Factory of the Future to Realise CIM*, Intertechno '90, Budapest, 1990, S. 515–524
- [21] GAJSKI, Daniel D.; KUHN, Robert H.: *Guest Editors' Introduction: New VLSI Tools*. In: *Computer*, Jg. 16, H. 12, 1983, S. 11–14
- [22] WALKER, Robert A.; THOMAS, Donald E.: *A Model of Design Representation and Synthesis*. In: *Proc. 22nd DAC*, 1985, S. 453–459
- [23] WEBER, Konrad: *Eine Konzeption zur Automatisierung stratifizierter Entwürfe digitaler Systeme basierend auf sukzessiver Detaillierung objektorientierter Modelle*. Dissertation, TU München, 1985
- [24] STACHOWIAK, Herbert: *Allgemeine Modelltheorie*. Springer-Verlag, Wien New York, 1973
- [25] STACHOWIAK, Herbert: *Erkenntnisstufen zum Systematischen Neopragmatismus und zur Allgemeines Modelltheorie*. In: *Modelle — Konstruktion der Wirklichkeit*, Hrsg: Stachowiak, Herbert, Wilhelm Fink Verlag, München, 1983, S. 87–146
- [26] QUBELDEY-CIRKEL, Klaus; WAHL, Michael G.: *What's on in the EDIF Test Are(n)a?*. In: *Proc. of the 4th European EDIF Forum*, Warrington, GB, 1990, S. 92–101
- [27] RENTSCH, T.: *Object-Oriented Programming*. SIGPLAN Notices, Jg. 17, H. 12, 1982
- [28] WAND, Yair: *A Proposal for a Formal Model of Objects*. In: *Object-Oriented Concepts, Databases, and Applications*, Hrsg.: Kim, Won; Lochowsky, Frederick H., Addison-Wesley Publishing Company, Reading, Mass., 1989, S. 537–559
- [29] MINSKY, Marvin Lee: *The Society of Mind*. Simon and Schuster, New York, 1986
- [30] PIAGET, Jean: *The Child's Conception of the World*. Routledge and Kegan Paul, London, 1964
- [31] NEWELL, Allen; SIMON, Herbert A.: *Computer Science as Empirical Inquiry: Symbols and Search*. In: *ACM Turing Award Lectures: The First Twenty Years*, Addison-Wesley Publishing Company, Reading, Mass., 1987, S. 287–313
- [32] BOOCH, Grady: *Object Oriented Design with Applications*. The Benjamin/Cummings Publishing Company, Redwood City, 1991
- [33] DANIELL, James Donald: *An Object Oriented Approach to CAD Tool Control*. Dissertation, Carnegie Mellon University, Pittsburgh, Research Report No. CMUCAD-89-37, 1989