

**Eine Diskussion der Problemkreise und  
Lösungswege bei der Realisierung eines  
dialoggesteuerten Expertensystems zur  
Auslegung primärgetakteter Schaltnetzteile**

Vom Fachbereich Elektrotechnik der  
Bergischen Universität – Gesamthochschule Wuppertal  
angenommene Dissertation zur Erlangung des akademischen Grades  
eines Doktor-Ingenieurs

von  
Diplom-Ingenieur  
Hans Groschwitz  
aus  
Marktredwitz



## **Vorwort**

Die Forschungsvereinigung Antriebstechnik (FVA) ist ein Teilbereich des Vereins Deutscher Maschinen- und Anlagenbauer (VDMA). Im Rahmen der FVA betreiben in erster Linie mittelständische Industrieunternehmen Gemeinschaftsforschung. Die Forschungsergebnisse stehen bevorzugt den Mitgliedsfirmen zur Verfügung. Im Jahr 1988 wurde von der FVA ein Forschungsvorhaben ausgeschrieben, mit dem Ziel, ein Expertensystem zu erschaffen. Dieses sollte in der Lage sein, einen projektierenden Ingenieur der Fachrichtung Maschinenbau bei der Auslegung frequenzumrichterbetriebener Drehfeldmotoren zu unterstützen. Um auch einen Informationsaustausch mittels Kennlinienverläufen und Zeichnungen zu ermöglichen, sollten interaktive grafische Darstellungsformen eingesetzt werden können. Die bevorzugte Ablaufumgebung war der IBM kompatible Personal-Computer (mit 80286 CPU), der in der Industrie bereits weit verbreitet war. Zu Beginn des Vorhabens Ende 1989 war die Leistungsfähigkeit der PCs noch zu gering, um die zu erwartenden Wissensmengen (und den damit einhergehenden Speicher- und Rechenleistungsbedarf) handhaben zu können. Geeignete Expertensystemschemen für PCs waren nicht verfügbar. Daher galt es, vorhandene Rechnersysteme in den einzelnen Firmen zu nutzen, um Neuinvestitionen zu vermeiden. Für CAD-Anwendungen standen zumeist Graphic-Workstations (unter dem Betriebssystem UNIX) zur Verfügung. Um sich in dem jungen Marktsegment KI-Produkte mit der Wahl einer Expertensystemschemen nicht von einem Softwarehersteller und seiner mittelfristigen Existenz abhängig zu machen, sollten die Quelltexte der Schale der FVA zur Verfügung stehen, um nötigenfalls selbst Portierungen auf unterschiedliche UNIX-Derivate vornehmen zu können.

Daraus ergab sich die Forderung nach einer leistungsfähigen Expertensystemschemen, die im Quelltext frei verfügbar sein sollte. Sie sollte grafikfähig sein und mußte die anfallenden Wissensmengen bewältigen können. Mit der Einschränkung, daß die Quelltexte frei verfügbar sein sollten, war die Auswahlmenge bereits sehr klein geworden. Meine Wahl fiel auf die hybride Expertensystemschemen Babylon-2 der Gesellschaft für Mathematik und Datenverarbeitung (GMD), da sie modular aufgebaut ist und erfolgreich auf unterschiedliche Umgebungen übertragen worden ist. Sie ist in der Sprache CommonLISP programmiert worden. Zugunsten dieser leichten Portierbarkeit verzichtet Babylon-2 auf eine grafische Mensch-Maschine-Schnittstelle

und beschränkt sich auf die Nutzung der Datenströme für Standardein- und Standardausgaben.

Um möglichst wenig Projektzeit für die eigentliche Wissens eingabe zu verlieren, galt es, ein Konzept zu entwickeln, mit dem unter Umgehung einer großen Überarbeitung der Quelltexte der Expertensystems chale diese Grafikfähigkeit ergänzt werden könnte. Die im vorgegebenen Softwareumfeld (UNIX-Graphic-Workstations mit OSF/Motif-Grafikbibliothek) verfügbaren Verfahren sind von mir untersucht und auf ihre Tauglichkeit hinsichtlich der Aufgabenstellung bewertet worden. Die Ergebnisse werden ausführlich in dieser Arbeit diskutiert. Diese Betrachtung bezieht auch die wachsenden Möglichkeiten der PCs mit ein.

Ende 1990 stellte das Softwarehaus VW-GEDAS, ein Unternehmen der Volkswagen-Gruppe, das Produkt *babylon-3* vor. Ausgehend von den Quelltexten der Expertensystems chale Babylon-2 ist dort eine komplette Neuprogrammierung unter der LISP-Implementierung Allegro CL sowie deren Erweiterungen des Sprachstandards vorgenommen worden. Dies verhindert die Nutzung von LISP-Interpretern anderer Anbieter. In der FVA wurde entschieden, daß für das laufende FVA-Vorhaben sowie für zwei weitere bei der FVA anstehende Projekte *babylon-3* zum Einsatz kommen sollte, da ein Synergieeffekt für die einzelnen Mitgliedsfirmen abzusehen war und ihnen die Möglichkeit des Langzeitsupports durch GEDAS zur Verfügung stand. Der Umstand, daß die Volkswagen AG Mitglied der FVA ist, spielte hierbei auch eine Rolle. Schließlich erwartete man - im Gegensatz zu einer an einer Hochschule entwickelten Lösung - eine im praktischen Einsatz stabilere und robustere Software. Damit wurde das Konzept der Grafikerweiterung für das laufende FVA-Vorhaben nicht mehr benötigt.

Am Lehrstuhl für elektrische Maschinen und Antriebe wurde mir die Möglichkeit eingeräumt, dieses Konzept davon unbenommen weiterzuentwickeln, um die Realisierbarkeit eines wissensbasierten Assistenten für die Auslegung primärgetakteter Schaltnetz teile zu untersuchen. Die Kosten eines Netz teils bestimmen sich zu einem großen Teil aus den Materialkosten der benötigten induktiven Bauelemente. Mit zunehmender Frequenz kann ein Transformator zur Übertragung einer bestimmten Leistung mit geringem Bauvolumen ausgelegt werden. Mit dem Bauvolumen sinkt auch der Einsatz an Magnetmaterial. Das heißt, daß durch Erhöhung der Übertragungsfrequenz Kosten eingespart werden können.

Das Magnetmaterial für derartige Mittelfrequenzübertrager wird in genormten Bauformen angeboten. Thermische Probleme, aber auch die endlichen Wickeldrahtquerschnitte begrenzen eine Reduzierung des Bauvolumens. Weitere nichtlineare Einflußgrößen sowie die Diskretisierung aufgrund der Verfügbarkeit der Bauelemente bewirken nicht stetige Lösungsräume, die miteinander hinsichtlich des Optimierungsziels konkurrieren. Daher ist auch für erfahrene Entwickler die Auslegung eines Schaltnetzteils zeitaufwendig. Stückzahlen entscheiden in der industriellen Praxis oft über den Grad der Annäherung an das Optimum.

Diese Arbeiten führten einerseits zu der grafischen dialogorientierten hybriden Expertensystemschale EMA-XPS, andererseits zu der Wissensbasis TRAFEX, die unter *babylon-3* und EMA-XPS gleichermaßen ablauffähig ist. Zur Realisierung der Abauffähigkeit von *babylon-3* Wissensbasen unter EMA-XPS mußten die von der VW-GEDAS eingebrachten Erweiterungen der Expertensystemschale abgebildet oder nachgebildet werden. Hierbei sind insbesondere die Erweiterungen der Inferenzmaschine von Babylon-2 (die Babylon-Query-Language, ein Datenbankanwendungen ähnlicher Zugriff auf das Faktenwissen, sowie ein spezieller Taskprozessor zur Kontrolle des Programmflusses dialogorientierter Sitzungen in Mehrfenstertechnik) zu nennen. Die Übersetzung von Wissenskonstrukten erfolgt mittels eines in den Wissensbasis-Ladevorgang integrierten Importfilters. Weiterhin mußten die grafischen Dialogelemente zur Kommunikation mit dem Benutzer angepaßt werden. EMA-XPS stellt grafische Editoren zur Wissenseingabe sowie ein Online-Hilfesystem zur Syntax der Wissensrepresentationsformalismen zur Verfügung.

EMA-XPS ist mehrsprachig angelegt. Derzeit werden die Sprachen Deutsch und Englisch unterstützt. Sprache und Farbgebung werden über X-Resource-Dateien kontrolliert. Die Ausgabe auf Farb- und monochromen Bildschirmen wird individuell unterstützt. Neben der Vollversion der Software, die auch alle Wissenseditoren erzeugt, steht eine die Rechnerbetriebsmittel schonendere Laufzeitversion ohne Wissenseingabemöglichkeiten bereit. die Auslagerung dieser X-Resource-Informationen in separate Dateien ermöglicht es beispielsweise, eine französischsprachige Version zu erstellen, ohne in den Quelltexten Änderungen vornehmen zu müssen. EMA-XPS ist mittlerweile auch erfolgreich auf leistungsfähige Intel-PC-Plattformen unter dem frei verfügbaren UNIX-Derivat Linux portiert wor-

den. Quelltexte und Dokumentation der Schale sind im World-Wide-Web (WWW) veröffentlicht unter:

<http://wmwap1.math.uni-wuppertal.de/EMA-XPS>

Mit Ausnahme von OSF/Motif ist EMA-XPS ein lizenzkostenfreies Produkt. Die Nutzungsrechte sind entsprechend der X-Window Copyright-Notiz (für das Netzwerkfenstersystem X11 und Babylon-2), der General Public License (GPL) der Free Software Foundation (für EMA-XPS, den C-Compiler gcc und den LISP-Interpreter clisp) und des Linux-Betriebssystems (größtenteils GPL) geregelt. Zur Zeit wird unter dem Projektnamen Lesstif die OSF/Motif-Grafikfunktionalität nachgebildet (ebenfalls GPL).

Angesichts der Rezession der letzten Jahre hatte man bei VW 1994 entschieden, kostenintensive und wenig gewinnversprechende Projekte nicht länger weiterzuführen. Dazu gehörte auch *babylon-3*. Nun bestand für die FVA Handlungsbedarf. Zur Sicherung der Ergebnisse aus drei laufenden Vorhaben mußte eine Ersatzschale beschafft werden. EMA-XPS bot hierfür gute Voraussetzungen. Mittlerweile stellt EMA-XPS v2.1.4 auch im industriellen Einsatz erfolgreich seine Robustheit und Stabilität unter Beweis.

Abschließend möchte ich nicht versäumen, meinen Dank auszudrücken. An erster Stelle möchte ich Herrn Prof. Dr.-Ing. J. Holtz nennen. Durch die Bereitstellung der nötigen Betriebsmittel in den Jahren an seinem Lehrstuhl, sowie die forschersche Freiheit, die er mir einräumte, ist das Entstehen dieser Arbeit erst ermöglicht worden. Herrn Prof. Dr.-Ing. J. Heidepriem danke ich für die Heidepriem fruchtbaren Diskussionen, die zum Gelingen dieser Arbeit beigetragen haben. Mein besonderer Dank gilt meinen ehemaligen Kollegen am Lehrstuhl, insbesondere Ashwin Khambadkone. Er gilt aber auch Michael Block, Frans Ceelen, Andreas Hennecke, Joachim Kromberg, Jan Legenhausen, Stephan Peters, Sven Schlums, Klaus Senf und Karsten Vossberg für ihren Einsatz im Rahmen ihrer Studien- und Diplomarbeiten zum Wohle des Vorhabens. Schließlich möchte ich meiner Frau danken - einfach für alles.

Bad Salzuflen, im Juni 1999

# **Eine Diskussion der Problemkreise und Lösungswege bei der Realisierung eines dialoggesteuerten Expertensystems zur Auslegung primärgetakteter Schaltnetzteile**

Vom Fachbereich Elektrotechnik der  
Bergischen Universität – Gesamthochschule Wuppertal  
angenommene Dissertation zur Erlangung des akademischen Grades  
eines Doktor-Ingenieurs

von  
Diplom-Ingenieur  
Hans Groschwitz  
aus  
Marktredwitz

Tag der mündlichen Prüfung: 18.6.1999

Referent: Prof. Dr.-Ing. J. Holtz  
Korreferent: Prof. Dr.-Ing. J. Heidepriem





## Inhaltsverzeichnis

0	Einleitung .....	1
0.1	Einführung in den Problemkreis .....	1
0.2	Ziel der Arbeit .....	4
1	Vorgehensweisen zur Auslegung von Mittelfrequenztransformatoren in primärgetakteten Schaltnetzteilen .....	6
1.1	Kenngrößen von Transformatoren .....	6
1.2	Netzteilprinzipien .....	10
1.2.1	Längsgerichtete Netzteile .....	10
1.2.2	Sekundärgetaktete Schaltnetzteile .....	11
1.2.3	Primärgetaktete Schaltnetzteile .....	12
1.3	Topologien primärgetakteter Schaltnetzteile .....	12
1.3.1	Die primärseitige Gleichrichterschaltung .....	13
1.3.2	Sperrwandlerschaltungen .....	14
1.3.3	Der Eintakt-Durchflußwandler .....	15
1.3.4	Durchflußwandlerbrückenschaltungen .....	17
1.4	Wechselwirkung mit den Halbleiterventilen .....	19
1.5	Betrachtungen zum Werkstoffeinsatz .....	20
1.6	Kernbauformen von Magnetmaterialien für Mittelfrequenz- transformatoren .....	22
1.6.1	Bauformen ferromagnetischer Metallkerne .....	22
1.6.2	Bauformen von Ferritkernen .....	24
1.6.3	Kernmaterialien ferromagnetischer Metallkerne .....	27
1.6.4	Kernmaterialien von Ferritkernen .....	28
1.7	Transformatorauslegung .....	29
1.7.1	Kernverluste ferromagnetischer Metallkerne .....	29
1.7.2	Kernverluste von Ferritkernen .....	32
1.7.3	Thermische Betrachtung von Ferritkernen .....	34
1.7.4	Thermische Betrachtung von Metallkernen .....	35
1.7.5	Maximale Leistungsübertragung .....	37
1.7.6	Auslegungsalgorithmen .....	40
1.7.6.1	Transformatorauslegung mittels geschlossener Gleichungen .....	40
1.7.6.2	Transformatorauslegung mittels Nomogrammen .....	45
1.8	Zusammenfassung .....	47

2	Möglichkeiten einer softwaretechnischen Umsetzung.....	48
2.1	Bedarf einer wissensbasierten Teilautomatisierung .....	48
2.1.1	Datenbanken.....	49
2.1.2	Objektorientierte Programmierung .....	50
2.1.3	Regelbasierte Systeme .....	51
2.1.4	Zwanggesteuerte Systeme .....	52
2.1.5	Weitere Techniken der künstlichen Intelligenz .....	52
2.1.6	Entscheidung zugunsten hybrider Expertensystemscha- len .....	53
2.2	Hybride Expertensystemscha- len .....	54
2.2.1	Komponenten eines Expertensystems.....	55
2.2.2	Arten von Wissen.....	58
2.2.3	Wissensrepräsentationsformen hybrider Expertensystemscha- len .....	59
2.2.3.1	Objektorientierung.....	59
2.2.3.2	Produktionsregeln.....	61
2.2.3.3	Restriktionsüberwachung über Constraintnetze.....	64
2.2.3.4	Weitere Wissensrepräsentationsformen .....	65
2.3	Auswahl einer geeigneten hybriden Expertensystemscha- le.....	66
2.4	Resourcenbedarf.....	68
2.5	Terminalanwendungen und ereignisorientierte Anwendungen in Mehrfenstertechnik.....	69
2.5.1	Interaktive Grafik.....	69
2.5.2	Programmfluß von Terminalanwendungen.....	71
2.5.3	Programmfluß ereignisorientierter Anwendungen .....	72
2.5.4	Ereignisorientierte Programmierung .....	73
2.5.5	Aufwandsabschätzung bei der Umstellung von Terminal- anwendungen auf Mehrfenstertechnik .....	76
2.6	Zusammenfassung .....	77
3	Nebenläufige Erweiterung von Terminalanwendungen um eine grafische Oberfläche am Beispiel von Babylon .....	79
3.1	Nebenläufigkeit in gängigen Betriebssystemumgebungen.....	79
3.1.1	Prozessorzuteilung.....	81
3.1.2	Speicherzuteilung.....	81
3.2	Möglichkeiten der Synchronisation und Kommunikation.....	82
3.3	Die Pseudoteletype-Gerätetreiber und die X11-Anwendung xterm..	83
3.4	Bestandsaufnahme für die Schale Babylon.....	86
3.4.1	Modularer Aufbau der Expertensystemscha- le Babylon.....	86
3.4.2	Die Mensch-Maschine-Schnittstelle von Babylon.....	87

3.5	Anforderungsprofil für eine Erweiterung der Schale Babylon.....	88
3.6	Umsetzung des vorgeschlagenen Lösungsansatzes unter UNIX-Betriebssystemen.....	90
3.6.1	Minimalversion der Gesamtanwendung.....	90
3.6.2	Ereignisorientierter Sitzungsablauf.....	92
3.6.3	Wechselnde Master-Slave-Beziehung.....	92
3.6.4	Interprozeßkommunikationskanäle.....	93
3.6.5	Post-Mortem-Verhalten.....	95
3.6.6	Unterbrechungen.....	95
3.6.7	Die Hauptschleife des LISP-Interpreters.....	96
3.6.8	Virtuelle Kanäle.....	96
3.6.9	Zustandsmaschine als Beobachter des LISP-Interpreters.....	98
3.7	Beispiel für die Verallgemeinerbarkeit des nebenläufigen Ansatzes.....	104
3.8	Zusammenfassung.....	106
4	Strukturierung und Implementierung des akquirierten Fachwissens.....	107
4.1	Betrachtung der Vorgehensweisen bei der Auslegung.....	107
4.2	Anforderungen an den Sitzungsablauf.....	110
4.3	Wissensstrukturierung.....	113
4.3.1	Objektorientierte Repräsentation des Faktenwissens.....	114
4.3.2	Constraintbasierte Repräsentation des Kontrollwissens.....	115
4.3.3	Auslegungsalgorithmen und Bewertungsfunktion.....	116
4.3.4	Regelbasierte Repräsentation des Ableitungswissens.....	118
4.3.4.1	Regelmenge zur Systembestimmung.....	119
4.3.4.2	Regelmenge zur Kernbestimmung.....	120
4.3.5	Benutzerschnittstelle mit grafischer Erklärungskomponente.....	123
4.4	Implementierung eines Expertensystems zur Auslegung des Mittelfrequenztransformators primärgetakteter Schaltnetzteile.....	124
4.4.1	Wissensstrukturierung.....	125
4.4.2	Realisierung einer Oberfläche mit grafischem Hilfesystem ..	126
4.4.3	Dokumentation des Sitzungsablaufs.....	128
4.5	Zusammenfassung.....	132
5	Schlußwort.....	133
6	Schrifttum.....	135



## **0. Einleitung**

### **0.1 Einführung in den Problemkreis**

Schaltnetzteile werden heute in der Stromversorgung vieler elektrischer Anlagen eingesetzt. Im Vergleich zu linear geregelten Netzteilen zeichnen sie sich durch geringere Verluste und somit Betriebskosten, sowie einem geringeren Einsatz an Magnetmaterial aus. Die Kosten eines Netzteils bestimmen sich zu einem großen Teil aus den Material- und Herstellungskosten der benötigten induktiven Bauelemente.

Mit zunehmender Frequenz kann das Bauvolumen eines Transformators zur Übertragung einer bestimmten Leistung reduziert werden. Mit dem Bauvolumen sinken der Einsatz an Magnetmaterial und Kupfer, wie auch der Wickelaufwand. Eine höhere Frequenz als die des speisenden Netzes läßt sich durch den Einsatz von Halbleiterschaltern erreichen. Mittels dieser Halbleiterschalter wird die zunächst gleichgerichtete Netzspannung hochfrequent wechselgerichtet. Das bedeutet, daß durch den Einsatz kostengünstiger Halbleiterbauelemente sowohl die Herstellungs-, als auch die Betriebskosten der Stromversorgung verringert werden können.

Die unterschiedlichen Magnetmaterialien für derartige Mittelfrequenzübertrager werden in einer Vielzahl genormter Bauformen angeboten. Aufgrund produktionstechnischer Abwägungen wie der Lagerhaltung kann diese Grundgesamtheit aber bereits firmenintern eingeschränkt worden sein. Bei der Optimierung hinsichtlich Kosten oder Baugröße hat dies Diskretisierungen und damit Unstetigkeiten der Lösungsräume zur Folge. Thermische Probleme bei der Wärmeabführung, aber auch die notwendigen Wickeldrahtquerschnitte begrenzen eine Reduzierung des Bauvolumens. Das gewählte Schaltnetzteilprinzip beeinflußt Art und Anzahl der Wicklungen. Weitere nichtlineare Einflußgrößen wie die Hysterese des Magnetmaterials schaffen Lösungsräume, die miteinander hinsichtlich des Optimierungsziels konkurrieren. Daher ist auch für erfahrene Entwickler die Auslegung eines Schaltnetzteils zeitaufwendig.

Die Stückzahlen entscheiden im industriellen Alltag oft über den Aufwand, der in die Entwicklung und damit die Optimierung eines Schaltnetzteils gesteckt wird. Es gilt also nicht, das absolute Optimum zu finden, um sich daraufhin einen geeigneten Kern anzufertigen. Vielmehr müssen die verfügbaren Komponenten optimal eingesetzt werden. Die Nichtlinearitäten der Kennlinien von Magnetmaterialien sind zumeist nur näherungsweise be-

kannt. Insbesondere sind entwicklungsbegleitend thermische Reihenuntersuchungen notwendig.

Die Hersteller der Magnetmaterialien bieten unterschiedliche Lösungswege zur Einsatzbestimmung ihrer Produkte an, die den Kunden, also den Schaltnetzteilentwickler, mehr oder weniger komfortabel an sein Ziel führen sollen. Dabei spielen Erfahrungen der dortigen Entwicklungsabteilungen eine Rolle, sowie die jeweilige Firmenpolitik, die Sicht auf die eigenen Kunden, aber auch das Bemühen, den Mitbewerbern am Markt geringstmöglichen Einblick in Firmengeheimnisse zu eröffnen. Dem Kunden wird ein Lösungsweg vorgegeben, der nur ein Mindestmaß an Transparenz gewährt. Beispielsweise werden Kennlinienverläufe als Tabellen angeboten, so daß Zwischenwerte interpoliert werden müssen. Andere Hersteller stellen geschlossene Näherungsformeln zur Verfügung. Teilweise existieren auch grafisch unterstützte Softwareanwendungen, die einen interaktiven Zugang zu diesen Näherungsformeln oder Tabellendaten bieten. Weiterhin bedeutet die Beschränkung auf die Daten, die für den empfohlenen Lösungsweg benötigt werden, daß die nichtgenannten Parameter Schwankungen unterliegen dürfen. Dies läßt höhere Exemplarstreuungen bei der Serienfertigung oder Modifikationen des Herstellungsprozesses zu.

Allgemein läßt sich feststellen, daß Datenblätter mit Parametern, die sich aus allgemeingültig herleitbaren physikalischen Formeln ergeben und die einen direkten Vergleich zuließen, eher selten zu finden sind [M1, M2].

Für den Entwickler, der eine Auswahl aus Produkten der unterschiedlichsten Hersteller vornehmen soll, bedeutet dies, daß er auf der Suche nach dem geeignetsten Kern oft auch den Auslegungsalgorithmus wechseln muß. Aufgrund der vorgenannten Diskretisierungen, die in Unstetigkeiten münden, und der Nichtlinearitäten ist der zeitaufwendige Auslegungsvorgang darüberhinaus über die Gesamtheit der zu untersuchenden Kerne und Schaltnetzteilprinzipien iterativ zu wiederholen.

Im Bemühen um reduzierte Entwicklungszeiten bei möglichst optimaler Auslegung des späteren Schaltnetzteils wird ein Werkzeug gesucht, das herstellerübergreifende Assistenz bei der Transformatorauslegung bieten kann. Um den Interessen der Benutzer gerecht zu werden, sollte dieses Werkzeug eine intuitiv erlern- und bedienbare Software sein. Diese ermöglicht es dem Entwickler, sich auf seine eigentliche Aufgabe zu konzentrieren. Er gibt zunächst die Eckdaten des Schaltnetzteils ein und trifft gegebenenfalls aufgrund seines Erfahrungswissens einige Vorentscheidungen. Die Maschine

erledigt die zeitraubenden Berechnungen und präsentiert Ergebnisse, aus denen der Entwickler dann geeignet auswählen kann.

Um Einarbeitungszeiten in das Werkzeug selbst gering zu halten, empfiehlt sich die Verwendung zeitgemäßer Mehrfenstertechnik, die es erlaubt, den Benutzer durch die notwendigen Eingaben zu führen, sowie Ausgaben auch in grafisch aufbereiteter Form bereitstellen zu können.

Andererseits sind mitunter neue Produkte und damit neue Auslegungsalgorithmen einzupflegen. Hierfür ist ein entsprechendes Maß an Programmierkenntnis unumgänglich. Oft fallen Aufgaben der System- und Softwarepflege einem Repräsentanten der firmeneigenen EDV-Abteilung zu, der nicht unbedingt Fachmann für die Auslegung von Schaltnetzteilen ist. Er kann daher nur in Abstimmung mit dem Entwickler als Experten und späteren Benutzer die Implementierung vornehmen. Da auch der EDV-Beauftragte vorwiegend mit anderen Aufgaben befaßt ist, sollte diese Wissenseingabeschnittstelle der Software ebenfalls intuitiv und in Mehrfenstertechnik realisiert sein.

Um die Kosten eines derartigen Werkzeuges akzeptabel zu halten, empfiehlt es sich schließlich

- zum einen eine Rechnerhardware zum Einsatz kommen zu lassen, die bereits vorhanden ist. Beispielsweise kann dies der vorhandene CAD-Arbeitsplatz für die Leiterplattenentwicklung sein, noch geeigneter wäre der eigene Arbeitsplatz-PC. Dies sind Systeme im unteren bis mittleren Leistungsbereich der Datentechnik.
- Zum anderen sollten die Softwareentwicklungskosten in einem wirtschaftlich vertretbaren Rahmen bleiben, so daß sich eine Investition rentiert. Hierbei spielen auch die Wahl des zugrundeliegenden Betriebssystems sowie der Entwicklungsumgebung bei der Kostenbetrachtung eine tragende Rolle. Schließlich können durch die Erweiterung vorhandener und im Einsatz bewährter Produkte aufwendige Neuentwicklungen vermieden werden.

Wenn auch einige Unternehmen für Teile ihrer Produktpalette bereits proprietäre interaktive Softwarelösungen entwickelt haben, sind diese zueinander zumeist bereits in der Zielsetzung inkompatibel und stellen damit nur eine Änderung des angebotenen Algorithmus, nicht aber eine tatsächliche Erleichterung dar, weil selbst diese Anwendungen in unterschiedlicher Weise jeweils mit den Daten des Anforderungsprofils zu befüllen wären. Selbst das Erstellen einer Software, die die Eingabedaten an die einzelnen Fertigprogramme jeweils geeignet aufbereitet weiterreicht und die Ergebnisse

wieder in Empfang nimmt, um sie dem Benutzer einheitlich zu präsentieren, scheitert am proprietären Ansatz. Dieser bietet im Regelfall keine gesicherten Schnittstellen. Daher kann nicht gewährleistet werden, daß die Rahmenanwendung auch bei Versionswechseln der unterschiedlichen Teilanwendungen noch einsatzbereit bliebe.

Vielmehr sollte eine Analyse der preisgegebenen Daten und Vorgehensweisen eine Software ermöglichen, die eine herstellerübergreifend einheitliche Bedienung bietet, einheitliche Resultate ermittelt und so vergleichende Bewertungen ermöglicht. Für den Kunden wäre daher ein etabliertes Werkzeug wünschenswert, welches es den anbietenden Firmen erlaubt, Erweiterungsmodule zu entwickeln, die leicht in die vorhandene Softwareumgebung eingebunden werden können. Dies läßt sich über festgelegte Programmierschnittstellen und Datenformate erreichen. Eine derartige Software stellt natürlich auch insofern eine Herausforderung für die Unternehmen dar, als ein direkterer Vergleich der eigenen Produkte mit denen der Wettbewerber möglich wird.

## **0.2 Ziel der Arbeit**

Das Ziel dieser Arbeit ist es, einen geeigneten softwaretechnischen Weg zu erarbeiten, der es ermöglicht, einem Entwickler bei der Auslegung primärgetakteter Schaltnetzteile Assistenz bei der Wahl des geeignetsten Transformators zu bieten.

Dazu werden zunächst im ersten Kapitel der Transformator in seinem Umfeld in der Gesamtanordnung Netzteil einschließlich der gegenseitigen Wechselwirkungen betrachtet. Daraufhin wird der Transformator selbst als eine Gesamtanordnung bestehend aus Kern, Spulenkörper und Wicklungen gesehen und diese Komponenten jeweils für sich, sowie als Bestandteil dieser Gesamtanordnung untersucht. Dabei wird dem Kern mit seiner Fülle an Materialparametern ein besonderes Gewicht zufallen. Es wird sich zeigen, daß eine Herleitung allgemeiner Gleichungen nicht praktikabel ist, da die Hersteller von Kernen die dafür notwendigen Parameter nicht in ausreichender Weise bereitstellen. Daher werden Vorgehensweisen zur Transformatorauslegung aufgezeigt, die von den Herstellern von Magnetkernen unterstützt werden und die demgemäß auch zu auswertbaren Ergebnissen führen. Somit steht das für eine erfolgreiche Auslegung notwendige Fachwissen bereit, das der Entwicklungsingenieur bei einer Auslegung ohne maschinelle Unterstützung verwendet hätte.



Bevor dieses Fachwissen zielgerichtet strukturiert werden kann, ist eine Analyse des zu verwendenden Softwarewerkzeuges geboten. Im Kapitel 2 werden zunächst Kandidaten von Werkzeugen vorgestellt und bewertet, um daraufhin die Vor- und Nachteile einer Entscheidung zugunsten einer Neuentwicklung, der Verwendung eines Fertigsystems sowie teilfertiger Lösungen zu diskutieren. Dabei wird sich herausstellen, daß Expertensysteme auf der Basis hybrider Expertensystemschalen der geeignetste Weg sind, die gestellte Aufgabe umzusetzen.

Daraufhin wird ein Verfahren vorgestellt, das eine komplette Neuentwicklung einer Expertensystemschale überflüssig macht. Darüberhinaus ist es geeignet, als konzeptionelle Basis softwaretechnischer Erweiterung zu dienen. Vorhandene bewährte Programme mit einer für alphanumerische Zeichen und Textterminals ausgelegten Mensch-Maschine-Schnittstelle können so mit geringem programmiertechnischem Aufwand auch in der Zeit grafischer Benutzerschnittstellen weiterverwendet werden.

Die Tragfähigkeit dieses Ansatzes wird in Kapitel 3 exemplarisch verdeutlicht, indem die vorhandene und für den Betrieb an Textterminals ausgelegte hybride Expertensystemschale Babylon um eine leistungsfähige grafische Mensch-Maschine-Schnittstelle erweitert wird.

In Kapitel 4 wird schließlich die Umsetzung der Vorgehensweisen und Rechenvorschriften zur Transformatorauslegung diskutiert. Nach einer Ziel festlegung wird das bereitgestellte Fachwissen zunächst allgemein den unterschiedlichen Wissensrepräsentationsformen hybrider Expertensystems schalen zugeordnet. Schließlich wird eine konkrete Implementierung unter Verwendung der in Kapitel 3 vorgestellten Schale mit ihren Möglichkeiten vorgestellt. Die Arbeit endet mit einer Demonstration der Leistungsfähigkeit des so entstandenen Expertensystems, das auch in der Lage ist, die eingangs aufgestellten hard- und softwareseitigen Anforderungen zu erfüllen.

## 1. Vorgehensweisen zur Auslegung von Mittelfrequenztransformatoren in primärgetakteten Schaltnetzteilen

Für das Verständnis der Aufgaben, die ein Softwarewerkzeug zur Assistenz eines Entwicklungsingenieurs bei der Auslegung des Mittelfrequenztransformators primärgetakteter Schaltnetzteile zu bewältigen hat, werden im folgenden die notwendigen Kenngrößen für die Transformatorberechnung bestimmt, sowie die gängigen Prinzipien galvanisch getrennter Netzteile vorgestellt. Daraus werden schließlich Algorithmen zur Auslegung optimierter Transformatoren abgeleitet, die sich in der Praxis bewährt haben.

### 1.1 Kenngrößen von Transformatoren

Aus dem Durchflutungsgesetz [M3, M4] läßt sich für eine Spule um einen toroidförmigen Ringkern die Durchflutung  $Q$  aus dem Produkt des Stroms  $I$  und der Anzahl  $N$  der Windungen der Spulenwicklung bestimmen. Da im Inneren eines Toroids aus homogenem Material die Feldstärke  $\vec{H}$  gleichbleibend ist und parallel zum betrachteten Wegstück  $d\vec{s}$  liegt, gilt

$$H l = N I \quad (1.1)$$

wobei  $l = 2\pi r$  die mittlere magnetische Weglänge und  $H$  die magnetische Feldstärke im Mittelpunkt des Toroidquerschnitts ist (Bild 1.1).

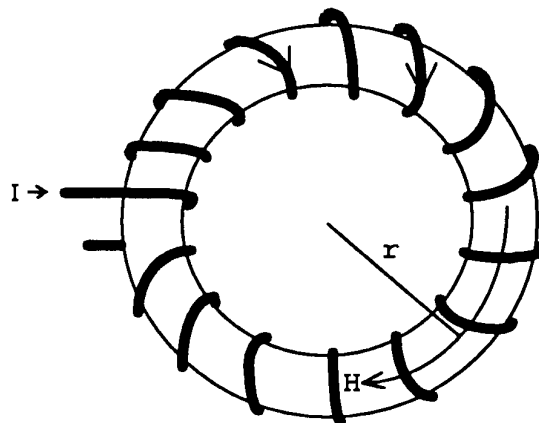


Bild 1.1: Toroidförmiger Ringkern

Durch die eindeutigen Zusammenhänge zwischen erzeugtem Magnetfeld und erzeugendem Strom bei einem Toroid werden die Eigenschaften der

Kernmaterialien mit toroidförmigen Kernen bestimmt. Die in der technischen Anwendung auftretenden Magnetkerne sind aufgrund des Wickelaufwands selten toroidförmig. Die Eigenschaften dieser Kerne werden dann entweder durch Näherungsformeln an die des Toroidkerns angenähert oder aus diesen abgeleitet. Mehr und mehr gehen allerdings die Hersteller dazu über, die Materialkenngrößen direkt für den jeweiligen Kerntyp anzugeben. Dies hat den Vorteil, daß die Näherungen entfallen und das Optimierungsergebnis besser wird.

Das Induktionsgesetz [M3, M4] besagt, daß die in einer Drahtschleife induzierte Spannung  $u$  durch die zeitliche Änderung des magnetischen Flusses  $F$  bestimmt ist, der mit der Schleife verkettet ist. Bei Spulen mit mehreren Windungen, die vom gleichen Fluß  $F$  durchflossen sind, gilt

$$u = N \frac{d\Phi}{dt} \quad \text{wobei} \quad \Phi = \vec{B} \cdot \vec{A} \quad (1.2)$$

für die in der Spule induzierte Spannung. Daraus ergibt sich, daß sich trotz gleichbleibenden Flusses  $F$  die Flußdichte oder Induktion  $\vec{B}$  in Teilvolumina des Kerns mit unterschiedlichen Querschnittsflächen  $\vec{A}$  ändert.

Weiterhin ergibt sich aus einer zeitlichen Änderung des Flusses eine solche der Induktion. Der **Induktionshub**  $DB$  ist dabei die Differenz zwischen größtem und kleinstem Wert der Induktion. Bei einem zeitlich sinusförmigen Verlauf gilt also  $DB = 2 \hat{B}$ , wobei  $\hat{B}$  die **Induktionsamplitude** ist.

Die hier betrachteten Anwendungen induzieren entsprechend Gleichung 1.1 das Magnetfeld im Kern durch einen im selben Spulendraht fließenden Strom. Dieser Vorgang heißt Selbstinduktion. Die in der Wicklung selbstinduzierte Spannung ist nach Gleichung 1.2 proportional zu  $dF/dt$ , der Änderungsgeschwindigkeit des Flusses. In isotropen Materialien ( $\vec{B}$  parallel  $\vec{H}$ ) kann mit dem Proportionalitätsfaktor der Induktivität  $L$

$$L = \mu \mu_0 N^2 \frac{A}{l} \quad (1.3)$$

der Zusammenhang zwischen Spannung und Strom

$$u = L \frac{di}{dt} \quad (1.4)$$

aufgestellt werden.

Voraussetzungen für die Gültigkeit der Gleichung 1.3 sind

- homogene Felder,
- konstanter Querschnitt des Kerns mit
- homogene Durchflutung und
- gleichbleibender Feldlinienlänge im Kern.

Diese Bedingungen werden von einem schlanken Ringkern mit im Verhältnis zum Ringinnenradius sehr kleinem Kernquerschnitt erfüllt.

In der Regel entsprechen die in der industriellen Fertigung eingesetzten Kerne diesen Voraussetzungen nicht. Für die verschiedenen Kernformen gleichen Querschnitts lassen sich die effektive Weglänge  $l_e$  und der effektive Querschnitt  $A_e$  durch ein Näherungsverfahren aus den inneren und äußeren Maßen so bestimmen, daß sie die Kerneigenschaften eines idealen Ringkerns der Abmessungen  $A_e$  und  $l_e$  beschreiben. Die Berechnung wird von den Herstellern der Kerne durchgeführt und ist in den IEC-Publikationen 205, 205A und 205B festgelegt. Durch die Normung ist so ein Standard entstanden, der herstellerübergreifende Vergleiche zuläßt. Bei dieser Berechnung wird ein kernformspezifischer Formfaktor benutzt, der in der Regel von den Herstellern angegeben wird.

Eine weitere Formkenngröße ist die effektive Permeabilität [M5]. Besteht der Kern nicht aus einem Stück, sondern aus zwei Hälften, verbleibt unweigerlich ein Restluftspalt  $d$  von etwa 8-10  $\mu\text{m}$ . Dadurch ergibt sich eine effektive Permeabilität  $m_e$ .

$$m_e = \frac{m_i}{1 + \frac{d \cdot m_i}{l_e}} \quad (1.5)$$

$d$  ist hier die Größe des Luftspalts und  $m_i$  die **Anfangspermeabilität**, also die Steigung der Neukurve im Nullpunkt der Hysterese Kennlinie. Gleichung 1.3 kann somit verallgemeinert werden zu

$$L = m_e m_0 N^2 \frac{A_e}{l_e} \quad (1.6)$$

oder vereinfacht

$$L = A_L N^2 \quad (1.7)$$

Der Induktivitätsfaktor  $A_L$  stellt also eine Zusammenfassung von Formkenn

größen dar (Tabelle 1.1). So wird eine einfache Berechnung der Spuleninduktivität ermöglicht.

Aufgrund der fertigungsbedingten individuellen Variation des verbliebenen Luftspalts der effektiven Permeabilität ergeben sich im Induktivitätsfaktor erhebliche Ungenauigkeiten. Kernhälften mit garantierten Toleranzen des  $A_L$ -Wertes werden deshalb nur paarweise geliefert. Die Toleranzen gelten dann nur für diese Paarung.

$l_e$	effektive magnetische Weglänge
$A_e$	effektiver magnetischer Querschnitt
$A_{\min}$	minimaler Kernquerschnitt, erforderlich zur Auslegung von $B_{\max}$
$V_e = l_e A_e$	effektives magnetisches Volumen
$\underline{m}_e$	effektive Permeabilität
$\Sigma (l/A) = l_e/A_e$	Formfaktor
$A_L$	Induktivitätsfaktor

Tabelle 1.1: Zusammenstellung wichtiger Formkenngrößen

Es ergeben sich einfache Zusammenhänge zur Berechnung der Scheitelwerte  $\hat{H}$  und  $\hat{B}$  aus den Scheitelwerten von Strom und Spannung bei sinusförmigem Spannungsverlauf am Transformator:

$$\hat{H} = \frac{\hat{I} \cdot N}{l_e}, \quad \hat{B} = \frac{\hat{U}}{2 \cdot \underline{p} \cdot f \cdot N \cdot A_e} \quad (1.8)$$

Bei rechteckförmiger Spannung ergibt die zeitliche Integration hingegen

$$\hat{B} = \frac{\hat{U}}{2 \cdot f \cdot N \cdot A_e} \quad (1.9)$$

Bei allen genannten Größen ist zu beachten, daß diese Werte nicht ohne weiteres von den statischen Meßbedingungen auf die in der Leistungselektronik geforderte dynamische Anwendung mit großen Aussteuerungen  $\hat{B}_{\max}$  übertragen werden können. Hat der Spulenkern keinen konstanten Querschnitt, ändert sich nämlich die Flußdichte in den verschiedenen stark durchfluteten Volumenelementen. Da die Permeabilität der Volumenelemente dann nicht mehr gleich ist, muß in diesem Fall die effektive Permeabilität durch eine Amplitudenpermeabilität  $\underline{m}_a$  ersetzt werden.  $\underline{m}_a$  ist abhängig von der angelegten Induktion, der Temperatur und der Frequenz. Die Fre-

quenzabhängigkeit tritt erst ab Schaltfrequenzen über 100 kHz in Erscheinung [M5]. Werte für  $m_b$  wurden erst in jüngster Zeit veröffentlicht.

$$m_a = \frac{\hat{B}}{m_0 \cdot \hat{H}} \quad (1.10)$$

Das wichtigste Kriterium zur Auslegung eines Kerns ist die im Transformator entstehende Verlustleistung  $P_V$ . Sie setzt sich zusammen aus den **Verlusten im Kern ( $P_K$ )** und den **Verlusten in den Wicklungsdrähten ( $P_N$ )**.  $P_K$  entsteht, wenn der Kern mit einer bestimmten Frequenz und einer bestimmten Induktionsamplitude ummagnetisiert wird.  $P_V$  erhöht die Temperatur des Kerns und muß nach außen abgegeben werden. Um die maximal zulässige Betriebstemperatur nicht zu überschreiten, darf  $P_V$  nicht höher werden, als es die Möglichkeiten der Gesamtanordnung Netzteil zur Wärmeabfuhr zulassen.

## 1.2 Netzteilprinzipien

Bevor detaillierter auf die Auslegung von Mittelfrequenztransformatoren eingegangen werden kann, ist eine Betrachtung des Umfeldes des auszuliegenden Transformators notwendig. Das Netzteilprinzip hat großen Einfluß auf die Wahl des Kerns und die Art und Form der Wicklungen.

### 1.2.1 Längsgergelte Netzteile

Die Stromversorgung elektronischer Geräte und Anlagen erfolgt in der Regel über Netzteile. Die Netzteile konventioneller Art bestehen aus einem Netztransformator, einem Gleichrichter, einem Ladekondensator, einer Siebdrossel und einem Siebkondensator. Der Transformator ist für die Netzfrequenz ausgelegt und besteht aus einem Blechpaket aus einer Eisen-Silizium-Legierung oder einer Eisen-Nickel-Legierung. Er sorgt zugleich für eine galvanische Trennung des Verbrauchers von dem speisenden elektrischen Netz.

Neben den aufgeführten Einzelteilen kann das Netzteil noch eine Stabilisierung enthalten. Damit ist es geeignet, innerhalb seines Arbeitsbereiches als annähernd ideale Strom- oder Spannungsquelle zu arbeiten. In der konventionellen Form besteht dieses Stabilisierteil aus einem Längs- oder Querregeltransistor und der dazugehörigen Regelschaltung. Die Netzteile haben beträchtliche Verluste im Transistor, weil dieser im linearen Arbeitsbereich als veränderlicher Widerstand eingesetzt wird. Da diese Verluste

vom Transformator mitgetragen werden müssen, sind die Netztransformatoren stabilisierter Netzteile konventioneller Bauart immer größer als die Transformatoren nichtstabilisierter Netzteile derselben Leistung. Durch die Verluste im Regelglied, das heißt, im Längs- oder Quertransistor, ergibt sich ein unbefriedigender Wirkungsgrad, besonders bei Vollast und Netzüberspannung.

Bei längsgeregelten Netzteilen ergibt sich die Verlustleistung aus der vom Längstransistor aufgenommenen Spannungsdifferenz und dem abgegebenen Strom des Netzteils. Diese muß durch Wärmeabfuhr am Kühlkörper des Längstransistors abgeführt werden. Diese konventionellen Netzteile werden durch Schaltnetzteile abgelöst [M8, M9].

### 1.2.2 Sekundärgetaktete Schaltnetzteile

Sekundärgetaktete Schaltnetzteile entsprechen netzseitig den längsgeregelten. Das Stellglied ist durch eine Gleichstrom-Gleichstrom-Wandlerschaltung ersetzt. Aufgrund ihrer unterschiedlichen Funktionsweise werden diese Wandlerschaltungen eingeteilt in Hochsetzsteller, Tiefsetzsteller oder Inverswandler [M10-M12]. Bild 1.1a zeigt das Schaltungsprinzip eines Inverswandlers.

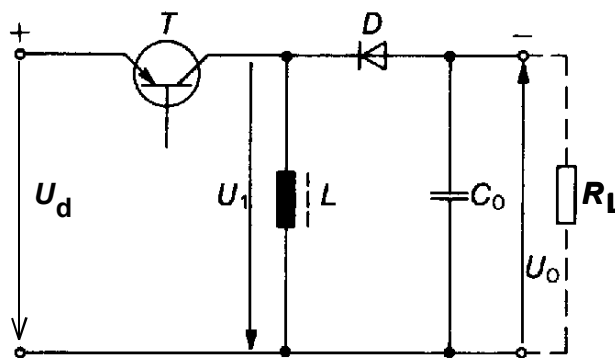


Bild 1.1a: Inverswandlerschaltung

Die sekundärgetakteten Schaltnetzteile weisen einen günstigeren Wirkungsgrad auf als die längsgeregelten, da die Transistoren als Schalter verwendet werden und in dieser Betriebsart verlustarm arbeiten. Dadurch kann der Netztransformator gegenüber den längsgeregelten Netzteilen etwas knapper dimensioniert werden. Die Transistoren werden in diesem Zusammenhang auch verallgemeinert als Halbleiterschalter oder Schaltventile bezeichnet. Die Anzahl der induktiven Bauelemente ist gegenüber den längsge-

regelten Netzteilen nicht geringer geworden. Dies gilt insbesondere für den Netztransformator, auf den nicht zuletzt wegen der geforderten galvanischen Trennung nicht verzichtet werden kann.

### 1.2.3 Primärgetaktete Schaltnetzteile

Die Übertragung von Wechselstrom geschieht in einem Transformator galvanisch entkoppelt. Die Änderung des primärseitigen elektrischen Stroms erzeugt einen magnetischen Fluß im Magnetmaterial. Dessen Änderung bewirkt eine Spannungsinduktion im Sekundärkreis und über die Last wieder einen Stromfluß. Die elektrische Energie  $W_{el} = U I t$  wird dabei in magnetische Energie  $W_m = \frac{1}{2} B H V$  überführt und sekundärseitig wieder in elektrische Energie zurückgewandelt. Bei steigender Frequenz werden die Leistungs-Zeit-Flächen kleiner. Damit sinkt der Bedarf an magnetischer Speicherkapazität. Das Volumen  $V$  kann verringert werden.

In primärgetakteten Schaltnetzteilen wird zunächst die Netzspannung mittels einer Diodenbrückenschaltung gleichgerichtet. Diese **primärseitige Gleichrichtspannung**  $U_d$  wird im Anschluß in eine Rechteckspannung im Frequenzbereich von etwa 1000 Hz bis etwa 200 kHz zerhackt, im Transformator galvanisch entkoppelt übertragen und sekundärseitig wieder gleichgerichtet. Die Regelung der Ausgangsspannung erfolgt entweder durch eine Veränderung der Arbeitstakt-Zeit bei konstanter Arbeitsfrequenz, oder durch Änderung der Frequenz bei konstanter Arbeitstakt-Dauer.

Wegen der hohen Arbeitsfrequenz können kleine Transformatoren mit Ferritkernen oder Kernen aus ferromagnetischen amorphen Metallegierungen verwendet werden. Der Einsatz an Magnetmaterial ist deutlich geringer als bei längsgeregelten Netzteilen. So können, obwohl der Aufwand an Halbleiterelementen höher ist, die Gesamtherstellungskosten des Netzteils gesenkt und der Wirkungsgrad sowie das Bauvolumen gegenüber längsgeregelten Netzteilen verbessert werden.

## 1.3 Topologien primärgetakteter Schaltnetzteile

Durch die Frequenzsteigerung wird ein kleineres Bauvolumen des Transformators möglich. Die Kernverluste der magnetischen Komponenten beschränken aber eine beliebige Verringerung der Kerngröße. Die Leistungsdimensionierung des Transformators eines primärgetakteten Schaltnetzteils wird in erster Linie durch die maximal erlaubte Temperatur im Wickelraum begrenzt. Sofern keine Vorkehrungen zur Kühlung getroffen sind, spielen die thermischen Eigenschaften des Transformators also eine große Rolle



hinsichtlich seiner Einsetzbarkeit. Schaltnetzteilentwickler sind vordringlich an einem handhabbaren Modell des Übertragers interessiert, das auch seine Umgebungseinflüsse berücksichtigt. Dabei handelt es sich um Größen wie die anliegenden Spannungen und Ströme, das netz- und lastseitige Verhalten, Angaben zur Effizienz, des Bauvolumens und der geforderten Güte.

Bei galvanisch entkoppelten Schaltnetzteilen wird die Netzspannung zunächst gleichgerichtet und mittels eines Kondensators stabilisiert. Ein anschließender Wechselrichter speist den Mittelfrequenztransformator primärseitig. Sekundärseitig wird erneut gleichgerichtet. Die fünf meistverwendeten Topologien sind [M10, M13, M14]:

- Sperrwandler,
- Sperrwandlerbrücke,
- Eintakt-Durchflußwandler,
- Durchflußwandler-Halbbrücke und
- Durchflußwandler-Vollbrücke.

Diese Topologien sollen im folgenden näher betrachtet werden.

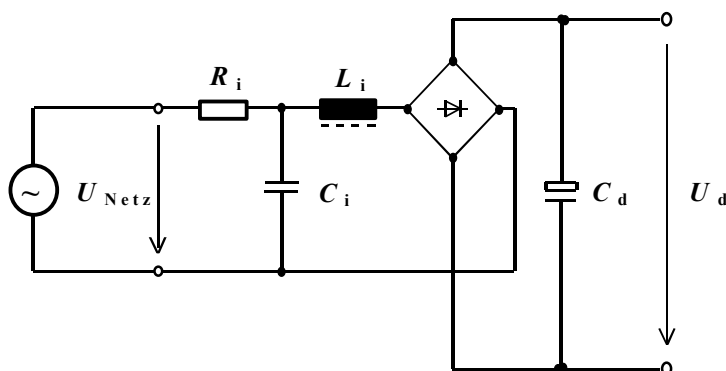


Bild 1.1b: Netzseitige Beschaltung primärgetakteter Schaltnetzteile

### 1.3.1 Die primärseitige Gleichrichterschaltung

Die primärseitige Gleichrichterspannung  $U_d$  wird durch Gleichrichten der Netzspannung gewonnen (Bild 1.1b). Der Ladekondensator  $C_d$  sollte so dimensioniert werden, daß einerseits die zulässige Restwelligkeit von  $U_d$  eingehalten werden kann. Dazu sollte er soviel Energie speichern, daß das Schaltnetzteil bei Nennbelastung und Ausfall von zwei aufeinanderfolgenden Netzhalbschwingungen noch funktionsfähig bleibt. Andererseits sollte der effektive Nennstrom durch den Kondensator bei Betriebstemperatur eingehalten werden. Andernfalls erwärmt sich der Elektrolyt unzulässig. Die Lebensdauer sinkt. Der netzspannungsseitige Schutzwiderstand  $R_i$  begrenzt

den Einschaltstromstoß des Ladekondensators  $C_d$ . Die Elektrizitätsversorgungsunternehmen legen Grenzen für Rückwirkungen auf das speisende Netz fest. Das aus  $L_i$  und  $C_i$  bestehende Entstörglied dämpft die vom Schalten der Halbleiterventile verursachten höherfrequenten Schwingungen [M8, M10, M12].

### 1.3.2 Sperrwandlerschaltungen

Das einfachste Wandlerprinzip der primärgetakteten Schaltnetzteile ist das Sperrwandlerprinzip. Es ist aus dem Inverswandler (Bild 1.1.a) weiterentwickelt worden. Um eine galvanische Entkoppelung zu erreichen, wird die im Querzweig des Inverswandlers liegende Speicherinduktivität mit einer zweiten Wicklung versehen. Der Transformator ist primärseitig mit dem speisenden Schaltungsteil mit dem Schalttransistor und sekundärseitig mit dem lastseitigen Schaltungsteil des Sperrwandlers mit der Sperrdiode verbunden. Den Aufbau verdeutlicht Bild 1.2.

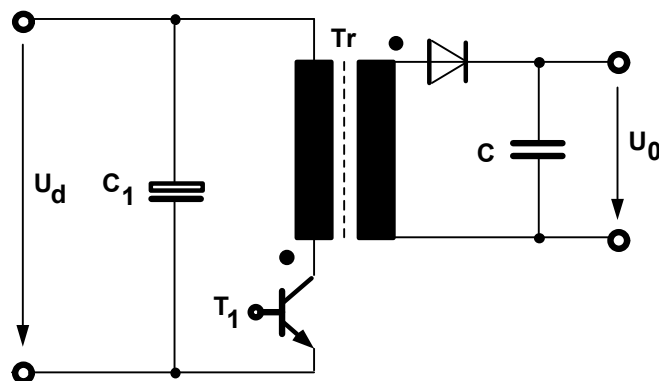


Bild 1.2: Grundsaltung eines Sperrwandlers mit primärer Tastung und Netztrennung

Wenn der Schalttransistor einen primärseitigen Strom zuläßt, nimmt der Transformator Energie auf. Diese Energie gibt der Transformator in der Sperrphase des Ventils sekundärseitig an die Last ab.

Im nichtlückenden Betrieb stellt sich für ein Tastverhältnis  $n$  die Ausgangsspannung

$$U_0 = \frac{n}{1-n} \frac{U_d}{ü} \quad (1.11a)$$

ein.

Sperrwandler eignen sich aufgrund ihres geringen Schaltungsaufwands besonders zum Bau preiswerter Schaltnetzteile im Leistungsbereich bis 200W, die nur geringe Güteanforderungen erfüllen müssen. Wird eine geringe Restwelligkeit von  $U_0$  gefordert, sollte auf andere Schaltnetzteilprinzipien zurückgegriffen werden. Die Streuung des Transformators ist aufgrund des zur Energiespeicherung benötigten Luftspalts relativ groß. Relativ zu anderen Wandlerprinzipien ist die elektromagnetische Beeinflussung der Umgebung hoch.

Die Sperrwandlerbrücke (Bild 1.3) zeichnet sich durch eine höhere primärseitige Spannungsfestigkeit aus. Sie sollte ab  $U_d > 1\text{kV}$  der Sperrwandlerschaltung bevorzugt werden.

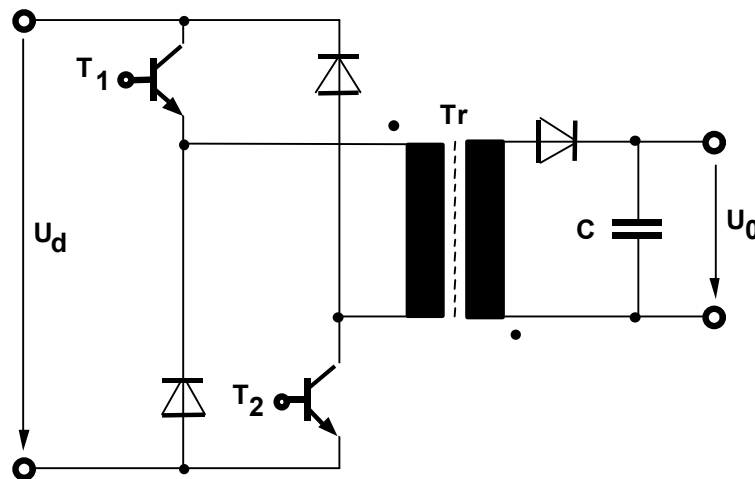


Bild 1.3: Sperrwandlerbrücke

### 1.3.3 Der Eintakt-Durchflußwandler

Für hochkompakt gebaute Netzteile und für Netzteile mit höheren Güteanforderungen oder Leistungsabgaben ist der Übergang zu Durchflußwandlern geboten. Der Eintakt-Durchflußwandler (Bild 1.4) kommt wie der Sperrwandler mit nur einem Schalttransistor aus.

Bei eingeschaltetem Transistor fließen primärseitig der durch den ausgangsseitigen Laststrom verursachte Strom  $I_0 \cdot i$  sowie der Magnetisierungsstrom durch die primärseitige Wicklung des Transformators. Aufgrund der Wicklungsorientierung ist die sekundärseitige Längsdiode leitend, der Stromkreis schließt sich über die Last.

Wird der Transistor abgeschaltet, ändert sich der Fluß  $F$ . Gemäß Gleichung 1.2 wird eine entgegengesetzt gerichtete Spannung in den Transfor-



Laststroms erreicht. Der primärseitige Wickelraumbedarf vergrößert sich um 20 bis 30 %. Für die Energieübertragung steht also nicht der volle Wickelraum des Kerns zu Verfügung.

Auch aufgrund der Eigenschaften der eingesetzten Halbleiterventile wird dieses Wandlerprinzip nur bis Übertragungsleistungen um etwa 800W verwendet [M10, M12].

- Der gesamte Schaltstrom wird über genau ein Ventil geschaltet. Demgemäß darf dieses im leitenden Zustand nur geringe Verluste aufweisen. Dies begrenzt den Einsatz von Feldeffekttransistoren. Sie sind zwar schnelle Schalter, haben aber hohe Verluste im leitenden Zustand.
- Zum anderen muß das Ventil auch eine hohe Sperrspannungsfestigkeit aufweisen. Bei bipolaren Transistoren erfordert wiederum die hohe Sperrspannung eine große Raumladungszone. Dies erhöht die Speicherzeit des Transistors und begrenzt dadurch seine maximale Schaltfrequenz [M15, M16].

### 1.3.4 Durchflußwandlerbrückenschaltungen

Bei Übertragungsleistungen oberhalb 800W ist es sinnvoll, die Leistung auf mehrere Transistoren in Gegentaktschaltung zu verteilen. Aufgrund der Gegentaktschaltung kann auch die Entmagnetisierungswicklung entfallen. Um den Transformator nicht in die Sättigung zu treiben, sollten die Ventile als Halbbrücke (Bild 1.5) oder Vollbrücke (Bild 1.6) angeordnet werden [M8, M10, M12].

Immer wenn einer der Transistorschalter des Gegentaktwandlers (Bild 1.5) leitet, ist auch eine der sekundärseitigen Dioden leitend. Bei ansteigendem Strom durch die Induktivität  $L$  wird dann sowohl Energie an die Last als auch an die Induktivität abgegeben. Während beide Ventile gesperrt sind, fließt die in der Induktivität gespeicherte Energie über die beiden Dioden zur Last.

Als Tastverhältnis wird die Einschaltdauer jeweils eines Ventils im Verhältnis zur gesamten Schaltperiode bezeichnet. Da pro Schaltperiode zwei Energieübertragungen stattfinden, verdoppelt sich am Netzteilausgang die Frequenz der Welligkeit von Strom und Spannung. Die typische Sperrspannung, mit der die Ventile beaufschlagt werden, beträgt  $U_d$ .

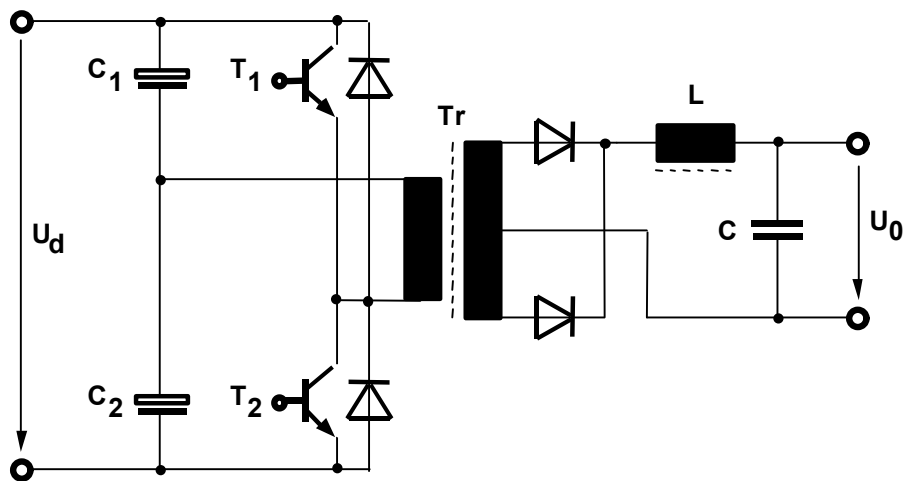


Bild 1.5: Durchflußwandler in Halbbrücken-Schaltung

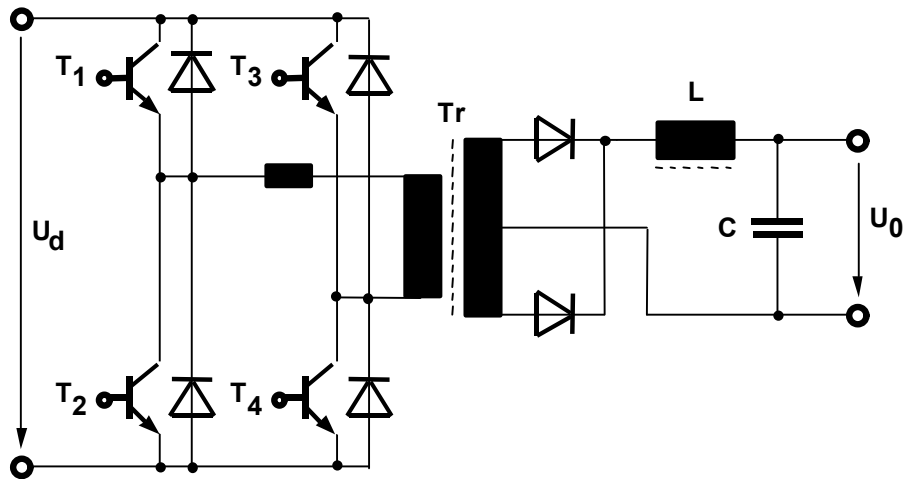


Bild 1.6: Durchflußwandler mit Vollbrücken-Primärkreis

Bei nichtlückendem Betrieb des Halbbrückenwandlers gilt für die Ausgangsspannung:

$$U_0 = n \frac{U_d}{\ddot{u}} \quad (1.11b)$$

Durch das Einschalten des jeweils anderen Ventils beziehungsweise Ventilpaars (in Bild 1.6 bilden  $T_1$  und  $T_4$  sowie  $T_2$  und  $T_3$  je ein Ventilpaar) wird die Spannung mit umgekehrtem Vorzeichen an den Transformator angelegt. Dadurch wird dieser einerseits wie beim Eintakt-Durchflußwandler entmagnetisiert, andererseits vollständig zwischen den beiden Maximalwerten der Induktion  $-B_{\max}$  und  $B_{\max}$  angesteuert. Durch diese Zentrierung des Induktionshubs auf den Nullpunkt der Flußdichte wird bei beidseitiger Vollauss-

steuerung der Induktion der Induktionshub verdoppelt. Gemäß Gleichung 1.2 kann der gleiche Fluß  $\Phi = \vec{B} \cdot \vec{A}$  bei doppeltem Induktionshub durch die halbe Querschnittsfläche transportiert werden. Bei Brückendurchflußwandler-schaltungen kann das Kernmaterial also wesentlich besser genutzt werden als bei den anderen Wandlerarten. Durch die Wahl eines kleineren Kerns sinken wiederum die Wicklungsverluste, da die Wicklungslängen abnehmen. Die Frequenz der Spannungswelligkeit an der Induktivität wird verdoppelt, die Amplitude der Restwelligkeit wird halbiert [M8, M11].

Andererseits ist bei Durchflußwandlerbrücken die Schaltung zum Ansteuern der Ventile erheblich umfangreicher als bei den bislang vorgestellten Schaltnetzteilprinzipien. Das ist ein wesentlicher Grund, der dem Einsatz von Halb- und Vollbrückendurchflußwandlern bei kleineren Leistungen entgegensteht. Die Ansteuerschaltung hat die Aufgabe, ein gleichzeitiges Durchschalten von zwei Ventilen einer Halbbrücke beziehungsweise zweier Ventilpaare einer Vollbrücke zu verhindern. Der Kurzschlußstrom würde die Ventile zerstören.

Rückblickend auf die vorgestellten Topologien und jeweiligen Randbetrachtungen erscheint die folgende Vorgehensweise bei der Auslegung von Schaltnetzteilen empfehlenswert:

- Zunächst wird die Schaltnetzteiltopologie unter Berücksichtigung der Leistungsabgabe und der Güte der Ausgangsspannung bestimmt.
- Im Anschluß daran sind die Leistungsschalter in Abhängigkeit von der angestrebten Leistungsbemessung und dem Wert der ausgangsseitigen Gleichspannung zu wählen. Die maximale Schaltfrequenz  $f_{\max}$  spielt bei der Auslegung eine entscheidende Rolle. Zusammen mit der Sperrspannungsfestigkeit bestimmt sie die Wahl der Leistungshalbleiterschalter.

#### **1.4 Wechselwirkung mit den Halbleiterventilen**

Die gegenseitige Beeinflussung von Schaltelementen und Transformator macht sich an erster Stelle im Rahmen der thermischen Leitung bemerkbar. Da die Kupferleitung zwischen Transistor und Transformator ein sehr guter Wärmeleiter ist, darf weder der Transformator wärmer werden, als die Transistoren es zulassen, noch umgekehrt. Das gleiche gilt für die sekundärseitigen Dioden [M15, M16]. Ferner ist auf eine geringe Streuinduktivität des Transformators zu achten, da sie zu Überspannungen an den Schaltventilen führt. Daher sind wechselseitig geschichtete Wicklungen empfehlenswert. Diese senken die Streuinduktivität der Wicklung um den Faktor 1,5 bis

3 [M8, M17]. Weiterhin ist darauf zu achten, daß die gesamte Stromführung von den Zwischenkreiskondensatoren über die Ventile und den Transformator bis zu den sekundärseitigen Gleichrichterdiolen induktivitätsarm aufgebaut wird. Alternativ kann ein Aufbau in Resonanztechnik erfolgen. Durch zusätzliche LC-Beschaltung wird die primärseitige Streuinduktivität des Transformators dabei zum Bestandteil eines Schwingkreises. Die Ventile müssen dann für doppelt hohe Spitzenwerte von Spannung oder Strom ausgelegt sein.

Die Bauelemente im Lastkreis bestimmen die Randbedingungen für die Transformatoroptimierung vollständig. Die Sperrfähigkeit der Ventile richtet sich nach der primärseitigen Gleichrichterspannung  $U_d$  (vgl. Bild 1.2), die maximale Schaltfrequenz nach deren Freierwerdezeit, der maximale Strom nach deren Schaltfähigkeit. Für die gewünschten elektrischen Ein- und Ausgangsgrößen und nichtelektrischen Parameter wie Größe, Gewicht, maximale Temperatur und die Kosten, die für das Gesamtsystem Netzteil gefordert werden, wird zunächst ein optimales Lösungsprinzip gesucht. Danach wird der Kern ausgelegt. Schließlich folgen die anderen Bauteile. Dabei wird davon ausgegangen, daß es einfacher und kostengünstiger ist, die restliche Schaltung an einen optimierten Transformator anzupassen.

Für die Funktionsweise des Transformators ist es von untergeordneter Bedeutung, welche Art von Schaltventilen oder Gleichrichterdiolen verwendet wird. Daher kann die Optimierung des Transformators getrennt von einer Optimierung der übrigen Schaltung im Schaltnetzteil durchgeführt werden [M18].

### **1.5 Betrachtungen zum Werkstoffeinsatz**

Zur Übertragung von Signalen wird der Transformator im Bereich niedriger Induktion betrieben, da dann die Übertragungseigenschaften annähernd linear sind. Die Leistungsübertrager arbeiten mit hoher Induktion und damit im nichtlinearen Bereich der Hystereseschleife. Beim Einsatz im Bereich hoher Induktionen ist man bemüht, die übertragene Leistung des Bauelements zu maximieren. Die Induktion ist dabei durch die Verluste im magnetischen Kreis nach oben begrenzt.

Im magnetischen Kreis kann ein Luftspalt vorgesehen werden. Dies ist bei Sperrwandlern der Fall, um die Energiespeicherfähigkeit zu erhöhen.

Die Reduzierung der Größe des Transformators durch eine Erhöhung der Schaltfrequenz wird bei Leistungsanwendungen durch die Erwärmung des Kerns begrenzt. Dies ergibt sich aus dem thermischen Widerstand des



Kerns und den Kern- und Wicklungsverlusten, sowie dem Ansteigen der Verluste mit der Schaltfrequenz. Gelingt es, die Verluste zu vermindern, kann ein kleineres Gesamtsystem, bestehend aus Transformator, Wicklung, Isolierung, Wickelkörper und Befestigungsmaterial, gewählt werden. Je effektiver das Kernmaterial gewählt wird, desto weniger Verluste werden im Gesamtsystem entstehen. Dadurch kann wiederum der Kern verkleinert werden. Andererseits sind bessere Materialien oft teurer, so daß sich eine solche Entscheidung wiederum nachteilig auf die Herstellungskosten des Netzteils auswirken kann.

Vergleicht man Werkstoffe unterschiedlicher Permeabilität, ist bei Einsatz des höherpermeablen Stoffes folgendes festzustellen:

- Im ungesättigten Bereich ist die Hysteresekennlinie steiler.
- Hohe Induktionen lassen sich mit niedrigeren Strömen oder geringeren Windungszahlen erzielen.
- Mit reduzierten Windungszahlen können die Wicklungsverluste verringert werden.
- Der gleiche Fluß kann bei höheren Induktionen durch geringere Kernquerschnitte geführt werden. Dies reduziert die minimal notwendigen Abmaße des Kerns und damit die Drahtlänge und die Wicklungsverluste.
- Die abnehmende Wickeldrahtlänge reduziert dessen Innenwiderstand. Die Sekundärspannung wird so laststabiler.
- Je kleiner der Kern ist, desto günstiger ist das Verhältnis von magnetischer effektiver Weglänge zum thermischen Widerstand des Kerns. Der kleinere Transformator kann so bezogen auf sein Kernvolumen spezifisch höher belastet werden. Das magnetisch relevante Volumen fällt mit der dritten Potenz der Kantenlänge, die thermisch relevante Oberfläche mit der zweiten Potenz.

Durch einen Arbeitspunkt des Transformators dicht unterhalb der Sättigung werden Netzüberspannungen scharf gekappt. Transformatoren verkraften gelegentliche Überspannungsspitzen großer Leistung. So schützen sie die sekundärseitigen Halbleiterschalter vor Überspannungen. Es genügt daher, die primärseitigen Ventile überspannungsfest auszulegen. Andererseits führt die Sättigung zu Überströmen, die wiederum die Ventile belasten. Eine Zerstörung der Schaltventile oder der Primärsicherung führt zum Ausfall des Netzteils. Daher sollte ein Betrieb zu nahe an der Sättigungsgrenze vermieden werden.

## 1.6 Kernbauformen und Magnetmaterialien für Mittelfrequenztransformatoren

Die in Kapitel 1.1 beschriebenen Transformatorgleichungen basieren auf Vereinfachungen, die sich aus Materialeigenschaften wie der Isotropie ableiten, sowie der Permeabilität. Eine nähere Betrachtung der erhältlichen Magnetmaterialien und ihrer Bauformen empfiehlt sich daher.

Die Bauformen der für Mittelfrequenztransformatoren geeigneten Kerne sind in verschiedenen Normen national und international festgelegt. Ferritkerne und Metallkerne werden dabei getrennt betrachtet. Die größte Vielfalt stellen die unterschiedlichen Ferritkerne dar. Während bei Metallkernen spezielle Bauformen auch bei kleineren Fertigungsserien denkbar sind, ist bei Ferritkernen ein Abweichen von den Standardbauformen bedingt durch den Herstellungsprozeß, einen volumenschwindenden keramischen Sinterungsprozeß, welcher jeweils eigene Pressformen benötigt, nur mit hohem Aufwand möglich und daher nur für große Stückzahlen wirtschaftlich.

### 1.6.1 Bauformen ferromagnetischer Metallkerne

Bei den ferromagnetischen Metallkernen unterscheidet man zwischen Ringkernen und Schnittbandkernen [M2, M19].

Der **Ringkern** ist hinsichtlich des magnetischen Kreises das optimale Bauteil. Anders als bei allen anderen Kernformen existieren keine weniger durchflossenen Bereiche. Mit dem Luftspalt entfällt allerdings auch die Möglichkeit einer einfachen Bewicklung unter Zuhilfenahme eines vorgefertigten Spulenkörpers. Der Einsatz eines Spulenkörpers setzt geteilte Kerne und damit die Existenz von Luftspalten voraus. Die Bewicklung ist sonst sehr kostspielig und erfordert Spezialmaschinen. Dazu kommen noch die Schwierigkeiten bei der Befestigung des gesamten Transformators. Ringkerntransformatoren bis zu mittleren Größen werden mit einer Schraube zentriert befestigt. Diese preßt den bewickelten Kern zwischen zwei Metall- und zwei Gummischeiben zusammen. Der Anpreßdruck wird vollständig auf die Drähte übertragen. Die Befestigung kann sich durch das Nachgeben der Isolation lockern. Größere Ringkerne werden daher immer in Becher eingegossen. Die Wärmeleitfähigkeit der Gußmasse und des Bechers sind dann ebenfalls zu berücksichtigen. Dies macht einen Ringkerntransformator zu einem sehr teuren Bauteil. Der Preis liegt gegenüber Lösungen mit Ferritkernen um den Faktor 3 höher. Sein Einsatz beschränkt sich daher auf Spezialgebiete.

Dieses Bauteil zeichnet sich durch sein sehr geringes Bauvolumen aus. Für die Größe eines Ringkerntransformators ist letztlich die Dicke der Wicklungen ausschlaggebend. Erfahrungsgemäß werden die Höhe und der Außendurchmesser bei voller Bewicklung um den Faktor 1,5 gegenüber den Abmessungen des Kerns steigen.

Ringkerne verursachen bei geeigneter maschineller Bewicklung die geringste elektromagnetische Beeinflussung der Umgebung. Sie haben sehr gute thermische Eigenschaften. Eine zusätzliche Abschirmung läßt sich bei vergossenen Übertragern einfach durch den äußeren Trog realisieren. Die Streuinduktivität wird alleine durch die Wicklung erzeugt. Für hohe Anforderungen an Gewicht, Streuarmut und Volumen stellen diese Kerne eine optimale Lösung dar.

Die wichtigsten Bauformen von Ringkernen sind in DIN 42311 und IEC 635 genormt worden. Die vielen Spezialanwendungen führen aber zu einem breit gefächerten Angebot. Es sind praktisch alle Variationen von Außendurchmesser, Innendurchmesser und Höhe erhältlich. Die Ringkerne werden entweder mit Harz vergossen oder in Schutztröge eingebettet, die teilweise auch schon die geforderten Isolierungen zwischen den Wicklungen enthalten können (Bild 1.7).

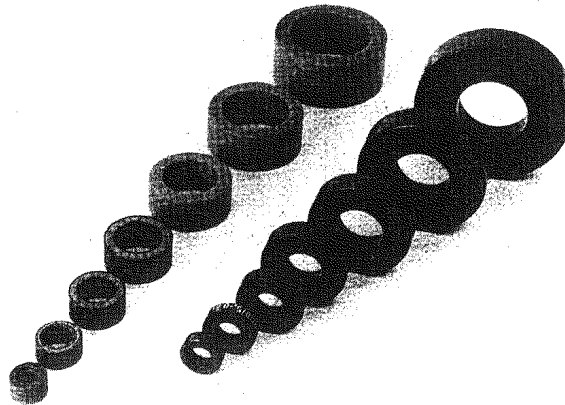


Bild 1.7: Ringkerne

**Schnittbandkerne** werden in der Regel als fertige Stücke bezogen, die paarweise zusammengehören. Die Paare sollten nicht vertauscht werden. Auch die Stellung der Hälften aufeinander ist farblich markiert. Nur so kann eine optimale Passung erreicht werden. Der Vorteil liegt in der Möglichkeit, vorgefertigte Spulenkörper verwenden zu können und trotzdem nur einen minimalen Luftspalt hinnehmen zu müssen. Wenn dennoch ein Luftspalt benötigt wird, läßt sich der Einfluß der Scherung [M31] genauer berechnen als

bei den geblechten Kernen von Netztransformatoren. Entscheidend ist hier das Verhältnis von Luftspaltlänge  $d$  zur Eisenweglänge  $l_e$ . Der wirksame Gesamtluftspalt liegt bei Schnittbandkernen zwischen 20 und 5  $\mu\text{m}$ , je nach Qualität des Schliffs. Damit liegt  $d/l_e$  im Bereich  $10^{-5}$  bis  $10^{-4}$ . Die resultierende Permeabilität lässt sich nach Gleichung 1.5 berechnen. Die Normung der Schnittbandkerne ist in DIN 41309 und IEC 329 erfolgt. Die Spulenkörper und Halterungen sind in DIN 41303 und DIN 41307-41310 festgeschrieben worden. Bild 1.8 zeigt die Bestandteile solcher Bausätze. Auf Anfrage können aber auch Sonderformen vom Hersteller geliefert werden.

Der Schnittbandkern aus amorphen Metallegierungen wird vor allem für die Anwendungen im Hochleistungs- und Hochspannungsbereich eingesetzt. Die Verlustleistung ist gegenüber Ringkernen um den Faktor 1,2 höher. Die Beschaffungskosten sind ebenfalls größer. Dies ist gegenüber dem höheren Wickelaufwand der Ringkerne abzuwägen. Die elektromagnetische Beeinflussung der Umgebung ist nur um wenige Prozent größer als bei Ringkernen. Die Streuinduktivität des Transformators ist bei mechanischer Bewicklung beider Kerntypen beim Schnittbandkern um den Faktor 1,3 größer.

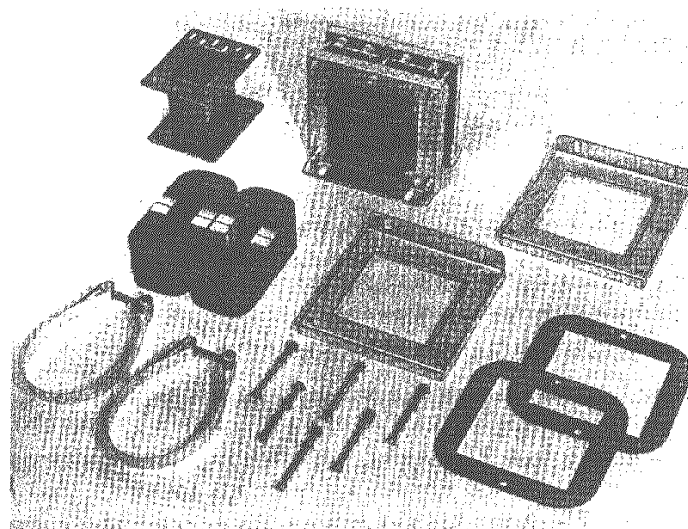


Bild 1.8: Schnittbandkerne

### 1.6.2 Bauformen von Ferritkernen

Ferritkerne werden derzeit in 14 verschiedenen Klassen von Bauformen gebaut. Es gibt folgende Kerntypen: E-, EC-, ED-, EFD-, EP-, ETD-, P-, PM-, Q-, R-, RM-, U- und UR-Kerne, sowie die veralteten und nicht mehr

angebotenen X-Kerne. Die Namensgebung ergibt sich häufig aus der Bauform des jeweiligen Kerntyps. Insbesondere bei den neueren Formen leiten sie sich aus der englischen Bezeichnung ab. Um eine - wenn auch unvollständige - Übersicht geben zu können, sollen im folgenden drei häufig verwendete Bauformen näher betrachtet werden [M1].

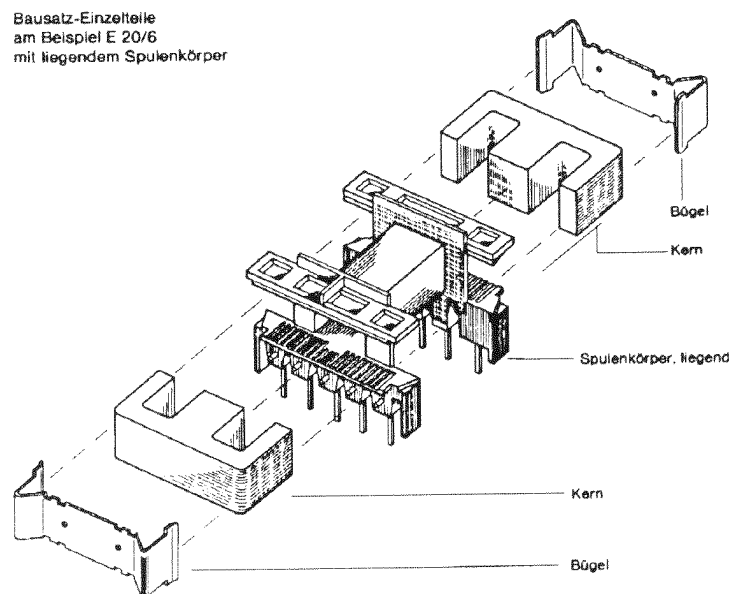


Bild 1.9: Ferritkern der Bauform E in liegender Montage mit Zubehörteilen

Bild 1.9 zeigt eine Explosionszeichnung der ältesten Bauform, des **E-Kerns**. Seine Bauform stammt von den genormten geblechten Transformatoren ab. Die äußeren Abmessungen sind durch DIN 41295 festgelegt. Für kleine Bauformen, früher auch als EF-Kerne bezeichnet, kam die DIN 41985 hinzu. In Kürze wird der IEC-Entwurf 51 (sec) 235 diesen Bereich erfassen. Die E-Kerne zeichnen sich durch große Wickelflächen aus und können liegend oder stehend eingesetzt werden. Die elektromagnetische Abstrahlung ist um den Faktor 4 größer als bei einem Schalenkern [M20].

Trotzdem haben sich die E-Kerne auch bei Mittelfrequenztransformatoren als Bauteil durchgesetzt. Die Einkopplung der elektromagnetischen Störungen in andere Bauteile kann durch liegende Montage bei dieser Kernform um den Faktor 1,5 verringert werden, da die Streufeldstärke dann nach oben und unten am größten ist, die Bauteile dagegen in einer Ebene mit den Ferritkernen liegen.

Der Schalenkern - englisch „pot core“ - wird als **P-Kern** bezeichnet (Bild 1.10). Da seine Spule von dem hochpermeablen Kernmaterial fast voll-

ständig umschlossen wird, hat diese Kernform von allen die geringste Streuinduktivität. Die elektromagnetische Abstrahlung der P-Kerne ist sehr gering. Bei Leistungsanwendungen liegt in dieser Geometrie aber auch der Nachteil dieser Kernform. Seine Wickelfenster sind sehr klein, und die Öffnungen für die Drahtanschlüsse begrenzen Querschnitt und Isolation erheblich. Die Wicklungsverlustwärme entsteht im Inneren und muß über das bereits durch die magnetischen Verluste erwärmte Kernmaterial nach außen abgeführt werden.

Die Parametrierung der P-Kerne erfolgt in Durchmesser mal Höhe. Die Schalenkerne sind in DIN 41293 und IEC 133 genormt. Erst seit wenigen Jahren existiert für einige Bauformen das Angebot automaten-gerechter Spulenkörper. Bis dahin mußten die Spulen nach der automa-tischen Bewicklung mit dem An-schlußträger per Hand verlötet werden, nachdem die Spule in den Kern eingesetzt wurde. Der Schalenkern entstammt der Nachrichtentechnik und wird auch heute noch meistens als Filterinduktivität oder Kleinsig-nalübertrager eingesetzt. Für Lei-stungsanwendungen kommt sein Einsatz nur bei Forderung nach be-sonderer Streufeldarmut in Betracht. Ansonsten ist er wegen der kleinen Wickelfenster wenig geeignet.

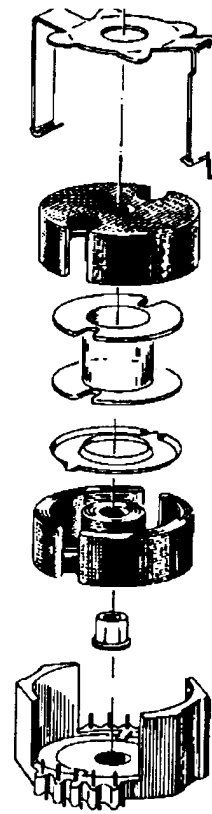


Bild 1.10: Explosionszeichnung eines Schalenkerns

Die neueste Bauform ist das „economic flat design“. Der **EFD-Kern** wurde für besonders flache Aufbauten entwickelt (Bild 1.11). Er ist lieferbar bis Baugröße 30. Dies entspricht der Länge seiner längsten Seite in Millimetern. Er ist für Energieübertragung bis 250 Watt bei Schaltfrequenzen bis 1 MHz geeignet.

Darüberhinaus ist er flußoptimiert. Die Flußoptimierung zeigt sich darin, daß der betrachtete Kern dieselbe Verlustleistung aufweist wie ein Ringkern mit äquivalentem magnetischen Querschnitt  $A_e$  und Kerngewicht  $m_K$ .

Dies bedeutet ein Minimum an Unsymmetrien in der magnetischen Durchflutung des Kerns. Die dazu nötige Berechnung wurde bei den neueren Kernformen durch Aufteilung in Teilvolumina gleichen magnetischen Querschnitts mit anschließender Berechnung der dort herrschenden Induktion und sich daraus ergebender Teilvolumenverlustleistung numerisch vorgenommen.

Der sich im Verhältnis zum Ringkern nach Summierung über die Teilvolumina ergebende Verlustleistungsfaktor ist ein Gütekriterium der Kernform. Es hat sich in der Praxis als sehr aussagekräftig erwiesen. Dieser Korrekturfaktor ist für die EFD-Kerne 1. Dadurch ergibt sich auch eine wesentlich geringere elektromagnetische Beeinflussung der Umgebung als bei E-Kernen. Gegenüber den Schalenkernen, die bei den diesbezüglichen Messungen als Referenz benutzt wurden, liegt sie nur um den Faktor 1,5 bis 3, je nach Lage, höher. Der EFD-Kern wurde für liegende Montage entwickelt. Die für seinen Einsatz nötigen Spulenkörper sind daher auch nur in liegender Bauform lieferbar.

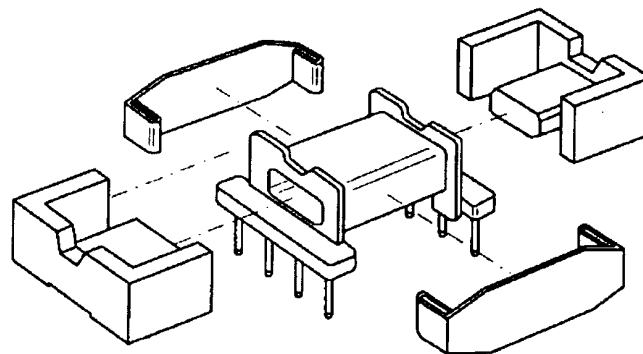


Bild 1.11: Explosionszeichnung des besonders für Flachbauweise geeigneten Kerntyps EFD

### 1.6.3 Kernmaterialien ferromagnetischer Metallkerne

Für den Einsatz in Mittelfrequenztransformatoren bietet sich nur die Reihe der amorphen Legierungen an. Nur sie haben in diesem Frequenzbereich vertretbare Verluste. Im Rahmen dieser Arbeit konnte mangels Verfügbarkeit nur die Reihe der VITROVAC<sup>®</sup>-Werkstoffe der Vakuumschmelze Hanau untersucht werden. Die verschiedenen Legierungen dieser Werkstoffreihe wurden jeweils auf spezielle Anwendungen hin optimiert. Für Schaltnetzteile nach dem Durchflußwandlerprinzip bei Übertragungsfre-

quenzen bis 150 kHz wird vom Hersteller das Material VITROVAC<sup>®</sup> 6030F empfohlen. Die nötigen Werkstoffwerte für die Optimierung des Transformators liegen vor [M2, M19, M21].

#### 1.6.4 Kernmaterialien von Ferritkernen

Bei Ferritkernen besteht die Möglichkeit einer breiteren Auswahl zwischen verschiedenen Werkstoffen unterschiedlicher Hersteller. Näher eingegangen wird hier auf die Werkstoffe der Firmen Siemens und Philips.

Die gängigen Leistungsübertragerwerkstoffe von **Siemens** werden mit N27, N47 und N67 bezeichnet. Der Werkstoff N27 ist das älteste Produkt. Er ist sehr preiswert und gut für das Frequenzgebiet bis 100 kHz geeignet. Darüber läßt seine Schirmwirkung nach [M20] und die Ummagnetisierungsverluste werden zu hoch. Man geht daher auf den Werkstoff N67 über, der bis 250 kHz empfohlen wird. Der relativ neue Werkstoff N47 ist für das Frequenzgebiet bis 1 MHz entwickelt worden. Ein genauer analytischer Zusammenhang zwischen Ummagnetisierungsverlusten, Induktionshub, Frequenz und Temperatur war von Siemens nicht erhältlich. Für diese Werkstoffe muß also aus den verfügbaren Diagrammen eine Funktion angenähert werden. Die Kernformabhängigkeiten bleiben bei allen von Siemens veröffentlichten Diagrammen unberücksichtigt.

**Philips** hat seinen Leistungsübertragerwerkstoffen die Kürzel 3C80, 3C85, 3F3 und 3F4 gegeben. Der Werkstoff 3C80 ist für den Frequenzbereich bis 100 kHz geeignet, daran schließt sich bis 200 kHz der Werkstoff 3C85 und bis 1 MHz das 3F3 Ferrit an. Für höhere Übertragungsfrequenzen kommt der Werkstoff 3F4 in Frage, der aber nur in wenigen Bauformen angeboten wird. Von Philips wurden für das 3F3 Ferrit Gleichungen zur Bestimmung der Ummagnetisierungsverluste für EFD-Kennformen in [M22] veröffentlicht. Bei Vergleich der Daten der Kernverluste bei gleicher Temperatur, Frequenz und Induktionsamplitude stellt man fest, daß 3F3 dem N47 von Siemens und 3C85 dem N67 entspricht. Über das Material 3C80 fehlen genaue Daten [M23-M27].

Neben den genannten Herstellern gibt es in Deutschland im wesentlichen noch die Firmen Vogt und Kaschke, die ein eigenes Kernprogramm vorhalten. Als ausländischer Hersteller ist die Firma Thomson in Deutschland vertreten. Auf den europäischen und amerikanischen Markt wurde hier nicht eingegangen, weil die Produkte dieser Hersteller bei hiesigen Distributoren nicht erhältlich sind.



## 1.7 Transformatorauslegung

Bei der Dimensionierung von Transformatoren für Schaltnetzteile stehen die Übertragung einer maximalen Leistung sowie minimale Verluste im Vordergrund. Die Verluste erwärmen den Kern und sind nach außen abzuführen. Die höchste zulässige Temperatur des Kerns darf nicht überschritten werden. Verluste entstehen sowohl im Kern als auch in den Wicklungen. Kernverluste bestehen zum einen aus Hystereseverlusten durch die Ummagnetisierung des Materials, zum anderen aus Wirbelstromverlusten.

### 1.7.1 Kernverluste ferromagnetischer Metallkerne

Bei Metallen hängen die Hystereseverluste  $P_H$  weniger von  $\hat{B}$  ab, denn die Sättigungsmagnetisierung liegt im Vergleich zu den Ferriten sehr hoch. Dafür ist aber der Anteil der Wirbelstromverluste  $P_W$  nicht zu vernachlässigen. Beide Anteile können getrennt werden, indem bei konstantem Induktionshub  $DB$  die Kernverluste pro Zyklus  $P_K/f$  über der Frequenz  $f$  aufgetragen werden. Die Kurve wird bis zu  $f = 0$  verlängert. Der Wert an dieser Stelle entspricht  $P_H/f$  und ist konstant. Der frequenzabhängige Teil der Kurve stellt  $P_W/f$  dar (Bild 1.12).

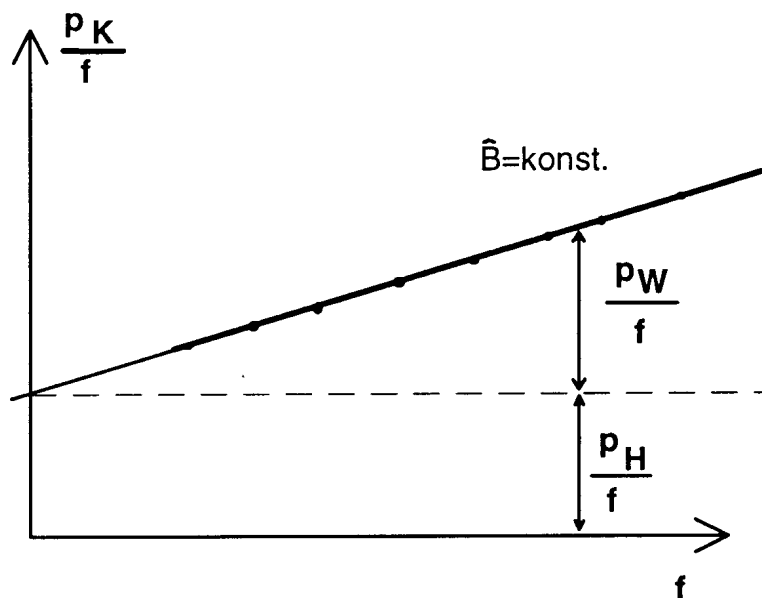


Bild 1.12: Separation der Verlustanteile in Ferromagnetika

Der Wirbelstromanteil der Verluste berechnet sich nach der Wirbelstromtheorie für Bleche [M3] pro Volumeneinheit zu

$$\frac{P_w}{V} = \frac{(\mathbf{p} \cdot d \cdot B \cdot f)^2}{6 \cdot \mathbf{r}} \quad (1.12)$$

wobei  $d$  die Blechdicke und  $\mathbf{r}$  der spezifische Widerstand des Blechs sind.

Bei amorphen Metallegierungen sind die grafisch aus Meßwerten ermittelten Werte für die Wirbelstromverluste annähernd gleich den nach der Gleichung 1.12 berechneten.

Bei der Mehrzahl der Materialien sind die gemessenen Verluste jedoch größer als die berechneten. Der Quotient aus gemessenen und berechneten Verlusten wird als Anomaliefaktor  $\mathbf{h}_w$  bezeichnet. Bei gewalzten Trafoblechen für die netzfrequente Energieübertragung kann  $\mathbf{h}_w$  Werte zwischen 2 und 3 annehmen. Der Grund für diese Abweichungen von der Wirbelstromtheorie liegt vor allem in der Annahme eines homogenen Materials in der theoretischen Betrachtung. Die Einflüsse der Weißchen Bezirke bleiben darin unberücksichtigt.

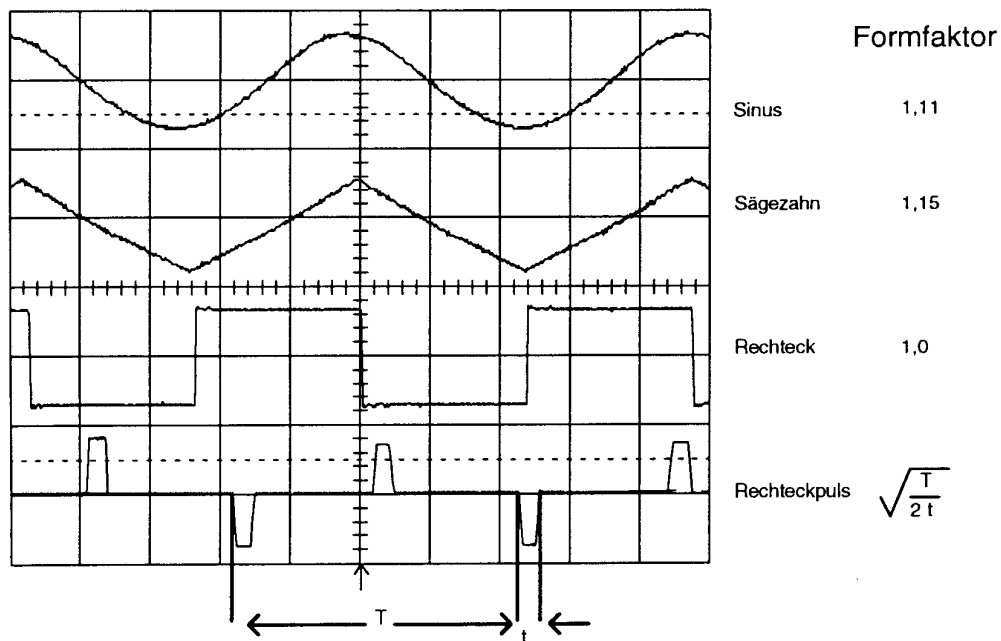


Bild 1.13: Verschiedene Spannungsformen mit den zugehörigen Werten der Formfaktoren  $F$  beziehungsweise  $F_0$

Die gewichtsspezifischen Kernverluste für Metallkerne ergeben sich nach [M28] allgemein zu:

$$\begin{aligned} p_K &= p_H + p_W \\ p_H &= c_1 \cdot \hat{B}^q \cdot f \\ p_W &= c_2 \cdot \hat{B}^v \cdot f^2 \end{aligned} \quad (1.13)$$

Dabei sind  $c_1$  und  $c_2$  Verlustbeiwerte,  $q$  und  $v$  Exponenten, für die gilt:  $q, v \in [1,3] \subset \mathbb{R}$ . Die Beiwerte und Exponenten werden teilweise von den Herstellern angegeben. Größtenteils müssen sie jedoch aus den Kurvenscharen der Hersteller ermittelt werden.

Der Faktor  $c_2$  ist stark von der Form der Spannung abhängig, mit der das Magnetmaterial ummagnetisiert wird. Meist wird er bei sinusförmiger Spannung bestimmt und muß für andere Spannungsformen umgerechnet werden. Man ersetzt daher  $c_2$  durch  $\underline{c}_2$  mit:

$$\underline{c}_2 = c_2 \frac{F^2}{F_0^2} \quad (1.14)$$

$F_0$  ist der Formfaktor der Meßspannung von  $c_2$ .  $F$  ist der Formfaktor der verwendeten Spannung (Bild 1.13).

Für die rechnerische Optimierung von Mittelfrequenztransformatoren auf minimale Verlustleistung wird jedoch der Übergang zu einer empirischen Verlustformel bevorzugt. Hierin sind beide Verlustanteile zusammengefaßt:

$$p_K = p_0 \cdot \left(\frac{F}{F_0}\right)^X \cdot \left(\frac{f}{f_0}\right)^Y \cdot \left(\frac{\Delta B}{\Delta B_0}\right)^Z \quad (1.15)$$

Sowohl die Bezugswerte  $p_0$ ,  $F_0$ ,  $f_0$  und  $\Delta B_0$ , als auch die Exponenten  $X$ ,  $Y$  und  $Z$  lassen sich aus dem in den Datenblättern des jeweiligen Kernmaterials gegebenen Kennlinienfeld der Verluste ermitteln. Da auch dies noch viel Rechenaufwand darstellt, begnügt man sich in der Praxis mit dem folgenden Verfahren:

1) Die verwendete Frequenz  $f$  mit Formfaktor  $F \neq F_0$  wird über

$$f_{\text{äquiv}} = f \cdot \frac{F^2}{F_0^2} \quad (1.16)$$

in eine äquivalente Frequenz umgerechnet.

- 2) Man entnimmt bei dieser Frequenz die Verluste bei gegebener Aussteuerung  $\hat{B}$  dem Kennlinienfeld des betreffenden Werkstoffs, gemessen mit Spannungsformfaktor  $F_0$ .
- 3) Diese abgelesenen Verluste werden mit

$$P_{K F_2} = \frac{F_1^2}{F_2^2} \cdot P_{K F_1} \quad (1.17)$$

auf die verwendete Spannungsform zurückgerechnet.

### 1.7.2 Kernverluste von Ferritkernen

Bei Ferritkernen haben die Wirbelstromverluste keine Bedeutung. Andere Verluste überwiegen hier. Ummagnetisierungsverluste entstehen vor allem aufgrund von thermischen Nachwirkungsverlusten, also temperatur- und frequenzabhängigen Aufweitungen der Hystereseschleife im Bereich der Koerzitivfeldstärke [M5]. Ein Maß für die Größe dieser Verluste ist die Fläche der Hystereseschleife. Von dieser Schleife geht auch die analytische Betrachtung der Verlustberechnung aus. Da es aber nicht möglich ist, von der mathematischen Beschreibung der Hystereseschleife bei Aussteuerung bis zur Sättigungsinduktion  $B_S$  zu einer geschlossenen Lösung der sich anschließenden Berechnung zu kommen, beschränkt man sich zunächst auf ein Gebiet, in dem die Hystereseschleife in einer einfachen Form mathematisch beschrieben werden kann. Ein solches Gebiet ist der Rayleighbereich. Er geht über den Bereich der Anfangspermeabilität hinaus, liegt aber meist noch unter der Koerzitivfeldstärke. Im Rayleighbereich ist die Hystereseschleife lanzettförmig und kann durch zwei Parabeläste angenähert werden (Bild 1.14). Als Gleichung für die Äste einer bis zur Feldstärke  $\hat{H}$  ausgerechneten Rayleighschleife erhält man:

$$B(H, dH) = \begin{cases} \mathbf{m}_0 \left[ (\mathbf{m}_i + 2\mathbf{V} \cdot \hat{H})h + \mathbf{V}(\hat{H}^2 - H^2) \right] \Big|_{dH \geq 0} \\ \mathbf{m}_0 \left[ (\mathbf{m}_i + 2\mathbf{V} \cdot \hat{H})h - \mathbf{V}(\hat{H}^2 - H^2) \right] \Big|_{dH < 0} \end{cases} \quad (1.18)$$

Die Konstante  $\mathbf{z}$  wird als Rayleighkonstante bezeichnet [M5].

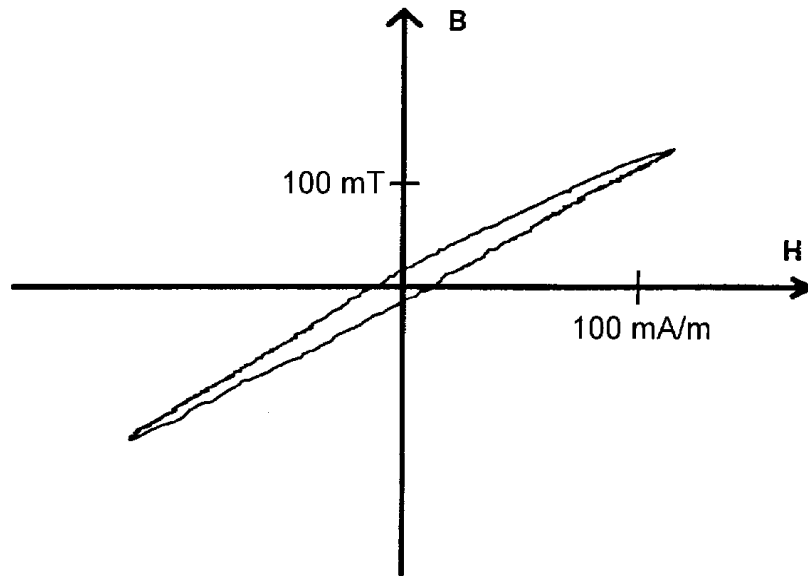


Bild 1.14: Lanzettform der Hystereseschleife im Rayleighbereich eines Ferritkerns E 25/7 aus dem Werkstoff N27 bei  $f = 20$  kHz

Beim Aufmagnetisieren des Werkstoffs wird Energie an das Kristallgitter abgegeben. Dadurch erwärmt sich das Material. Wird die Hysteresekennlinie  $B(H, dH)$  aus Gleichung 1.18 zunächst von  $-\hat{H}$  bis  $\hat{H}$  mit  $dH > 0$  durchlaufen und von dort wieder zurück mit  $dH < 0$  bis  $-\hat{H}$ , ergibt sich die Hystereseschleife als ein geschlossener Integrationsweg. Die Hystereseverluste, also die dabei in Wärme umgewandelte Energie, entsprechen der Fläche der Hystereseschleife.

$$w_H = \oint_{[-\hat{H}, \hat{H}], [\hat{H}, -\hat{H}]} B(H, dH) dH = \frac{8}{3} \mathbf{m}_0 \cdot \mathbf{V} \cdot \hat{H}^3 \quad (1.19)$$

Die Ummagnetisierungsleistung beträgt demnach für einen Kern mit Volumen  $V$  und Masse  $m_K$  bei der Frequenz  $f$ :

$$P_H = \frac{8}{3} \mathbf{m}_0 \cdot \mathbf{V} \cdot \hat{H}^3 \cdot f \cdot V \quad (1.20)$$

Da  $\mathbf{m}$  sich im Rayleighgebiet nur wenig ändert, gilt mit Gleichung 1.20:

$$\frac{P_H}{m_K} = p_H \sim f \cdot \hat{B}^3 \quad (1.21)$$

Außerhalb dieses Bereichs gilt allgemein:

$$\frac{P_H}{m_K} = p_H \sim f^n \cdot \hat{B}^m \quad (1.22)$$

Bei Mangan-Zink-Ferriten ist  $n \in [1,2] \subset \mathbb{R}$  und  $m \in [2,3] \subset \mathbb{R}$ . Um mit dieser Gleichung arbeiten zu können, wären genaue Herstellerangaben zu  $n$  und  $m$  nötig. Andernfalls sind diese nur mit hohem mathematischem Aufwand aus veröffentlichten Kennlinien zu ermitteln.

### 1.7.3 Thermische Betrachtung von Ferritkernen

Die im Kerninneren entstehenden Verluste müssen in Form von Wärmeenergie nach außen abgeführt werden. Vor allem bei Ferriten ist die thermische Leitfähigkeit des Kernmaterials gering [M9]. Außerdem ist der thermische Widerstand stark von der Bauform der Ferritkerne abhängig. Die Weglänge des Wärmestroms im Kernmaterial hat dort entscheidenden Einfluß auf das Gesamtsystem. Zwei Wege zur Ermittlung des thermischen Widerstands eines Kerns werden beschrieben. Der erste ist in [M10] aufgezeigt und gilt nur für Ferritkerne. Danach ergibt sich der thermische Widerstand eines Ferritkerns näherungsweise zu:

$$R_{th} = \frac{1}{A} \left( \frac{1}{\mathbf{a}} + \frac{g}{\mathbf{I}} \right) \quad (1.23)$$

mit:

$A$  Oberfläche des Bauteils

$g$  kleinste Totalabmessung des Bauteils

$\mathbf{a}$  Wärmeübertragungsfaktor der Kernoberfläche. Bei freier Konvektion ohne Kühlkörper und ohne Verguß für Ferrite gilt:

$$\mathbf{a} = 0,029 \cdot 10^{-3} \frac{W}{K mm^2}$$

$\mathbf{I}$  Wärmeleitungsfaktor im Kernmaterial. Für Ferrite gilt:

$$\mathbf{I} = 0,35 \cdot 10^{-3} \frac{W}{K mm}$$

Nachteil dieses Verfahrens ist, daß zuerst die Werte für  $A$  und  $g$  berechnet werden müssen. Durch die Näherungen ist es darüber hinaus ungenau.

Ein zweiter Weg besteht in der experimentellen Bestimmung dieses Wertes. In [M22] wurden Messungen für eine Reihe von Ferritkernen unter-

schiedlicher E-Kernformen durchgeführt. Es wurde ein Bezug gesucht zwischen dem thermischen Widerstand und einem oder mehreren elektromagnetischen Parametern. Eine gute Korrelation ergab sich bei dem Bezug auf das effektive Kernvolumen. Es wurde folgender Zusammenhang gefunden:

$$R_{th} = C_{th} \cdot V_e^{-0,54} \quad (1.24)$$

$C_{th}$  ist dabei von der Geometrie des effektiven Kernvolumens  $V_e$  und der Art der Bewicklung abhängig.

Leider existieren keine derart exakten Untersuchungsergebnisse über andere Kernformen, wenn auch Richtwerte für einige andere Kernformen in [M10] angegeben sind. Für die verbleibenden Kernformen empfiehlt sich eine experimentelle Bestimmung des thermischen Widerstands.

Das folgende Meßprinzip hat sich dabei bewährt: Ein Thermoelement wird zwischen die Berührflächen beider Kernhälften eingebracht. Der Kern wird so mit einer Spule bewickelt, daß der gesamte Wickelraum ausgefüllt wird. Die Spule wird mit Gleichstrom gespeist. Der thermische Widerstand des Kerns ergibt sich zu:

$$R_{th} = \frac{\Delta J}{U \cdot I} \quad \text{wobei} \quad \Delta J = J_{Kern} - J_{Raum} \quad (1.25)$$

Die Curietemperatur des Kerns darf dabei nicht überschritten werden.

Bei den im Rahmen dieser Arbeit durchgeführten Messungen zeigte sich, daß die in [M22] und [M10] angegebenen Werte nicht werkstoffabhängig sind. Dagegen besteht eine starke Abhängigkeit von der Dicke der Bewicklung. Je weniger Bewicklungslagen aufgebracht werden, desto geringer ist auch der thermische Widerstand des Kerns. Daher bezieht sich  $R_{th}$  immer auf die Gesamtanordnung Kern, Spulenkörper und Wicklung. Auch in den hier zitierten Veröffentlichungen [M1-M30] wird immer von einer vollen Bewicklung der Wickelfenster ausgegangen.

#### 1.7.4 Thermische Betrachtung von Metallkernen

Metallkerne können nicht nach dem oben vorgestellten Verfahren gemessen werden. Es besteht keine Möglichkeit, im Kerninneren einen Temperaturmeßpunkt zur Messung der Kerntemperatur anzubringen, ohne den Kern zu zerstören. So muß entweder auf die Herstellerangaben zurückgegriffen oder ein anderes Meßverfahren angewendet werden. Herstellerangaben sind im allgemeinen nur schwer zugänglich.

Ringkerngröße Abmaße in mm	thermischer Widerstand in K/W
16/10/6	39
20/12,5/8	27
25/16/10	18
30/20/15	13
40/25/15	8,4
50/40/20	4,7

Tabelle 1.2: Thermischer Widerstand von Ringkernen aus VITROVAC<sup>®</sup> 6030F ohne Schutztrog mit Epoxyfixierung

Kernmaterialspezifische Werte sind für die VITROVAC<sup>®</sup> Produktreihe (Tabelle 1.2) in [M2] veröffentlicht worden. Leider standen keine Angaben über weitere Kernmaterialien zur Verfügung. Nach Herstellerangaben schwanken die Werte aber nicht stark hinsichtlich unterschiedlicher Werkstoffklassen. Alle amorphen Metallegierungen weisen demnach diese thermischen Widerstände auf.

Für andere Kernabmessungen gibt es die Möglichkeit einer Messung. Wird ein bewickelter Kern auf die Kernaußentemperatur  $T_a$  erwärmt, und die Wärmequelle dann abgeschaltet, bei der obigen Meßschaltung ist dies die Gleichspannungsversorgung, fällt die Temperatur an der Außenseite des Kerns exponentiell ab mit

$$T(t) = T_a \cdot e^{-\frac{t}{\tau}} \quad \text{wobei } \tau = m_K \cdot c_p \cdot R_{th,K} \quad (1.26)$$

Der Temperaturanstieg ist festzuhalten. Aus dem Temperaturverlauf läßt sich dann die Zeitkonstante  $\tau$  ermitteln.

Zusätzlich benötigt wird noch die spezifische Wärmekapazität  $c_p$  des Kernmaterials. Diese kann entweder einem Vergleichswerk über Werkstoffe entnommen oder durch eine Vergleichsmessung mit einem Metallkern bekannten thermischen Widerstands ermittelt werden. Die Metallegierungen hochpermeabler Werkstoffe basieren meist auf Kobalt. Dem Kobalt werden etwa 20-30% Silizium zugesetzt. Die spezifische Wärmekapazität beträgt für Kobalt in dem hier relevanten Temperaturbereich



bei Erwärmung von 25-100°C

$$c_p = 0,1186 \frac{W \cdot h}{kg \cdot K}$$

und bei Erwärmung bis 200°C

$$c_p = 0,1221 \frac{W \cdot h}{kg \cdot K}$$

Für Silizium liegt sie in der Größenordnung

$$c_p = 0,1953 \frac{W \cdot h}{kg \cdot K}$$

Die einzelnen Wärmekapazitäten sind entsprechend der Legierung zu gewichten. Da die Legierung nicht bekannt ist, liegt in dieser Abschätzung eine große Unsicherheit dieses Meßverfahrens, wenn kein Vergleichskern vorhanden ist.

Die gängigen Kerngrößen sind in Tabelle 1.2 bereits erfaßt. Messungen beschränken sich daher in der Praxis auf Sonderaufbauten.

Abschließend sei bemerkt, daß der thermische Widerstand von Metallkernen ungefähr um den Faktor 5 geringer ist als der von Ferritkernen.

### 1.7.5 Maximale Leistungsübertragung

Eine wichtige Kenngröße für die Bemessung der übertragbaren Leistung ist die zulässige Temperaturerhöhung. Aus dieser wird über den thermischen Widerstand die maximale Verlustleistung von Kern und Spule berechnet.

Die Leistung

$$P_V = \frac{\Delta J_K}{R_{th,K}} \quad (1.27)$$

kann als Wärme über den Kern nach außen abgegeben werden. Diese setzt sich zum einen aus der im Kern entstehenden Wärme, zum anderen aus dem Teil der Wicklungsverluste zusammen, die über das Kernmaterial abgeleitet werden. Daher muß je nach Bauform abgeschätzt werden, welcher Anteil dieser Wicklungsverluste direkt an die Umgebung abgegeben wird und nicht zur Erwärmung des Kerns beiträgt. Bei PM-Kernen beträgt der Faktor der direktkonvektierenden Kupferverluste nur etwa 0,25, bei EF-Kernen und Ringkernen liegt er um 0,7. Maximale Erwärmung tritt dann ein, wenn alle Verluste durch den Kern abgeleitet werden müssen.

Für die weiteren Berechnungen ist das Verhältnis zwischen Kupfer und Kernverlusten sowie dessen Beziehung zum Minimum der Gesamtverluste zu betrachten. Dazu wird zunächst die Grundschialtung Bild 1.15 betrachtet.

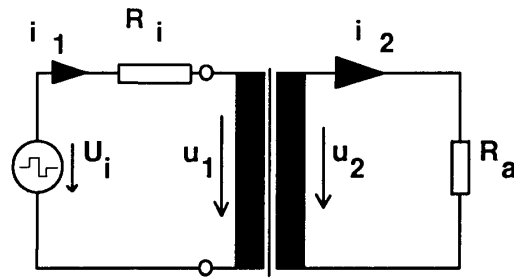


Bild 1.15: Grundschtung eines Mittelfrequenztransformators

$U_i$  und  $R_i$  repräsentieren den primärseitigen Wechselrichter. Dabei ist  $U_i$  eine zeitabhängige Funktion der Schaltfrequenz  $f$ , des Tastverhältnisses  $n$ , und der primärseitigen Gleichrichterspannung  $U_d$ . Wenn die in den Lastwiderstand  $R_a$  übertragene Leistung und das Übersetzungsverhältnis  $\ddot{u}$  vorgegeben sind, dann liegen auch die Ströme und Spannungen fest. Erhöht man die Windungszahl  $N_1$ , ist auch  $N_2$  zu erhöhen, da  $N_2 = N_1/\ddot{u}$ . Dadurch nimmt gemäß Gleichung 1.8 die magnetische Flußdichte oder Induktion ab. Die Kernverluste nehmen entsprechend den Gleichungen 1.20 und 1.22 ebenfalls ab. Der Wickelraum ist beschränkt, daher muß der Kupferquerschnitt der Wicklungen verkleinert werden. Demgemäß erhöhen sich die Kupferverluste  $P_N$ . Für  $P_K$  und  $P_N$  ergibt sich also eine gegenläufige Abhängigkeit von der Windungszahl sowie von  $\hat{B}$ . Für die Gesamtverluste  $P_V$  ergibt sich ein Minimum bei einer bestimmten Windungszahl. Dieses Minimum kann aber nur dann ausgenutzt werden, wenn die zum Optimalwert  $N_{1,opt}$  gehörende Induktionsamplitude unter der zulässigen Induktionsamplitude

$$\hat{B}_{zul} = \frac{A_{\min}}{A_e} \hat{B}_{\max} \quad (1.28)$$

liegt.  $\hat{B}_{\max}$  stellt die für den Kernwerkstoff maximal zulässige Betriebsinduktion dar. Sie wird für Leistungsübertragerwerkstoffe von den Herstellern angegeben. Diese Überprüfung gemäß Gleichung 1.28 ist nach jedem Optimierungsversuch erneut durchzuführen. Entsprechend Gleichung 1.22 verhalten sich die Kernverluste proportional zur Induktion mit der Potenz  $m$ , also  $\hat{B}^m$ , wobei  $m \in [2,3] \subset \mathbb{R}$  ist. Damit ergibt sich auch eine Abhängigkeit von der Windungszahl mit negativem Exponenten gleicher Größe.

$$P_K \sim \frac{1}{N_2^m} = \frac{\ddot{u}^m}{N_1^m} \quad (1.29)$$

Auch die Wicklungsverluste  $P_N$  sind von der Windungszahl abhängig. Werden ineinandergeschachtelte oder Kammerwicklungen vorausgesetzt, so haben beide Wicklungen die gleiche mittlere Weglänge  $l_N$ , die auch eine Vollwicklung hätte. Die verschiedenen Wicklungsanordnungen sind in Bild 1.16 dargestellt.

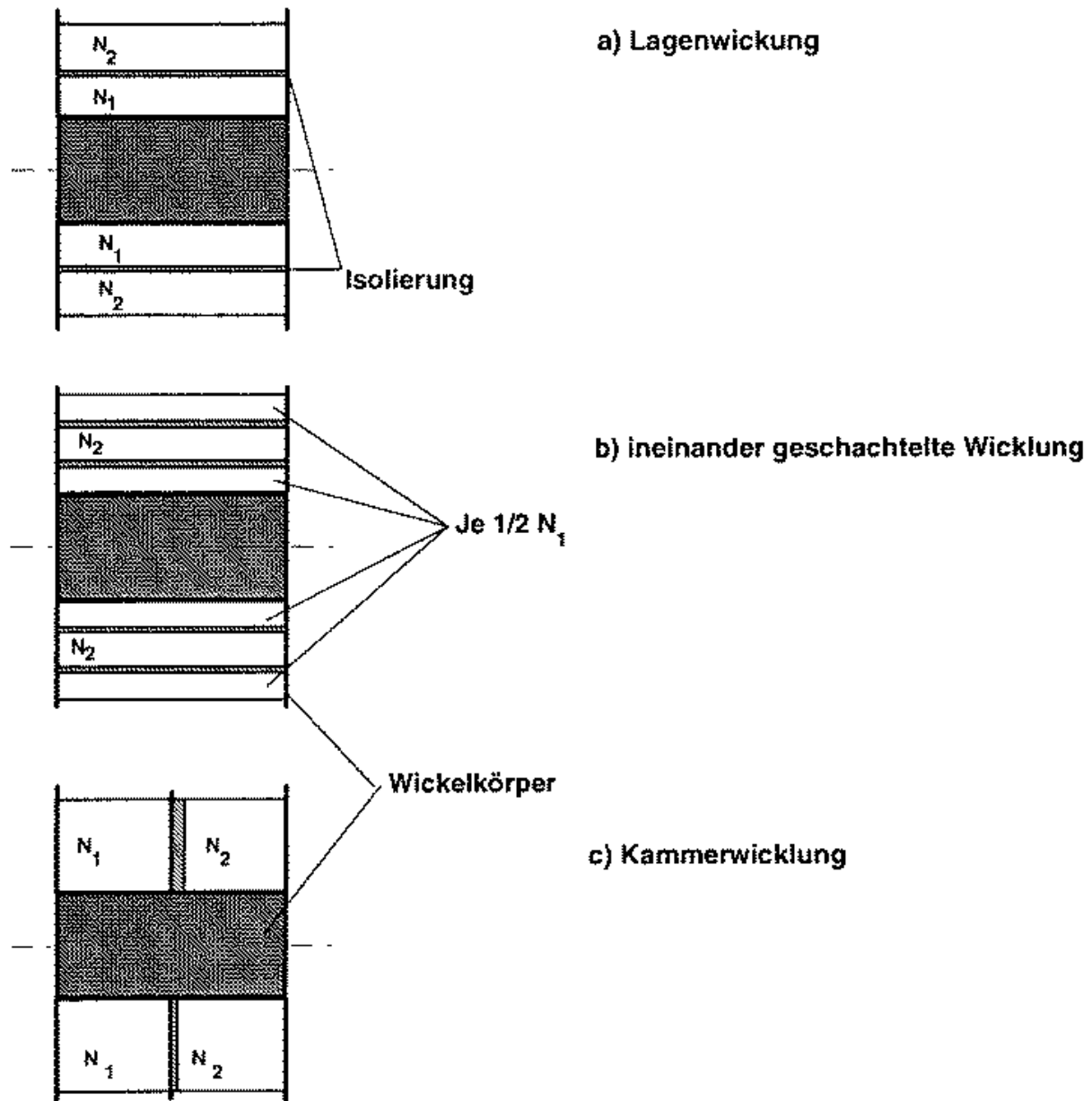


Bild 1.16: Unterschiedliche Wicklungsanordnungen von Primär- und Sekundärwicklungen

Aufgrund geometrischer Überlegungen ergeben sich die Wicklungswiderstände

$$R_{N1} = \mathbf{r}_{\text{Draht}} \frac{N_1^2 \cdot l_N}{\mathbf{e}_{N1} \cdot A_N} \quad (1.30)$$

für die Primär-, beziehungsweise

$$R_{N2} = \mathbf{r}_{\text{Draht}} \frac{N_2^2 \cdot l_N}{\mathbf{e}_{N2} \cdot A_N} \quad (1.31)$$

für die Sekundärwicklung aus dem spezifischen Widerstand des Drahtmaterials  $\mathbf{r}_{\text{Draht}}$ , seiner Länge  $l_N$  und seinem Drahtquerschnitt  $A_{\text{Draht},1} = A_N \mathbf{e}_{N1} / N$ .  $\mathbf{e}_{N1}$  und  $\mathbf{e}_{N2}$  bezeichnen die auf den ganzen Wickelraumquerschnitt  $A_N$  bezogenen Füllfaktoren.

Wie in [M5] hergeleitet, sind die Wicklungsverluste minimal bei einer Wickelraumaufteilung, die

$$\mathbf{e}_{N1} = \mathbf{e}_{N2} = \frac{1}{2} \mathbf{e}_N \quad (1.32)$$

erreicht.  $\mathbf{e}_N$  ist dabei der Füllfaktor, den man bei gleicher Raumausnutzung und gleicher Materialdichte für die Aufbringung nur einer einzelnen Wicklung im Wickelraum erhalten würde. Die Wicklungswiderstände verhalten sich proportional  $N^2$ .

Wird bei abnehmendem Drahtquerschnitt die Windungszahl geeignet erhöht, bleibt der Wicklungsquerschnitt konstant. Durch die geänderte Bepackung nimmt  $\mathbf{e}_N$  dagegen ab. Auch der Wärmetransport in der Wicklung außerhalb des Kupfers verändert sich. Die thermische Leitfähigkeit der Wicklung sinkt mit  $\mathbf{e}_N$ . Die durch die Wicklungsverluste bewirkte Erwärmung der Wicklung ist umgekehrt proportional zu  $\mathbf{e}_N$ . Damit nimmt die Wicklungsverlustleistung  $P_N$  etwas überproportional zu  $N^2$  zu.

Aufgrund der annähernd gleichen Werte der Exponenten liegt das Minimum der Gesamtverluste bei  $P_N \approx P_K$  [M5].

## 1.7.6 Auslegungsalgorithmen

### 1.7.6.1 Transformatorauslegung mittels geschlossener Gleichungen

In der Literatur werden drei praktikable Wege zur Optimierung von Kern und Wicklung angeboten [M10, M22]. Zwei davon wurden vor allem

für Ferritkerne, die dritte bevorzugt für Metallkerne entwickelt. Die prinzipiellen Unterschiede sind gering, problematisch bei der praktischen Anwendung ist die Verfügbarkeit der zur Optimierung nötigen Stoffparameter. Diese werden von den Herstellern jeweils nur für den von ihnen empfohlenen Berechnungsweg angegeben. Die Lösungswege verfahren zunächst einheitlich. Dieser Teil soll zunächst nachvollzogen werden.

Der hier dargestellte Weg wurde von Philips [M22] publiziert. Er geht zunächst von der Leistung aus, die der Transformator in einem Eintakt-Durchflußwandler liefert. Diese Leistungsabgabe des Netzteils kann durch Multiplikation des Ausgangsstroms mit der Ausgangsspannung dargestellt werden:

$$P_{\text{Übertragen}} = U_0 I_0 \quad (1.34)$$

Die Ausgangsgleichspannung  $U_0$  des Schaltnetzteils ergibt sich dabei in Abhängigkeit von dem Transformatorübersetzungsverhältnis  $\ddot{u}$ , der primärseitigen Eingangsspannung  $U_d$  und dem Tastverhältnis  $\mathbf{n}$ .

$$U_0 = \frac{\mathbf{n} U_d}{\ddot{u}} \quad (1.35)$$

Der Ausgangsstrom wird in [M22] in Abhängigkeit vom Strom durch die primärseitigen Transformatorwicklung mit

$$I_0 = \frac{I_1 \cdot \ddot{u}}{\sqrt{\mathbf{n}_{0,\max}}} \quad (1.36)$$

angegeben, wobei  $\mathbf{n}_{0,\max}$  das maximale Tastverhältnis im stationären Betrieb ist. Weiterhin gilt für die Wicklungsverluste:

$$P_N = I_1^2 (R_{N1} + \ddot{u}^2 \cdot R_{N2}) \quad (1.37)$$

Durch Umformung von Gleichung 1.37 nach  $I_1$ , anschließendes Einsetzen in Gleichung 1.36 und Einfügen des Ergebnisses unter Berücksichtigung von Gleichung 1.35 in Gleichung 1.34 ergibt sich:

$$P_{\text{Übertragen}} = U_d \cdot \mathbf{n} \cdot \sqrt{\frac{P_N}{\mathbf{n}_{0,\max} (R_{N1} + \ddot{u}^2 \cdot R_{N2})}} \quad (1.38)$$

Bezeichnet man das Verhältnis von Wechselstrom- zu Gleichstromwiderstand einer Wicklung mit  $\mathbf{g}_N$  und berechnet die Gleichstromwiderstände der primären und der sekundären Wicklung nach Gleichung 1.30 beziehungsweise 1.31, ergibt sich:

$$(R_{N1} + \ddot{u}^2 \cdot R_{N2}) = N_1^2 \cdot l_N \cdot \mathbf{r}_{\text{Draht}, T_{\max}} \cdot \left( \frac{\mathbf{g}_{N1}}{\mathbf{e}_{N1} \cdot A_N} + \frac{\mathbf{g}_{N2}}{\mathbf{e}_{N2} \cdot A_N} \right) \quad (1.39)$$

Ferner gilt gemäß Gleichung 1.9:

$$N_1 = \frac{\hat{U}_i}{2 \cdot A_{\min} \cdot \hat{B} \cdot f} \quad (1.40)$$

Dabei wurde der effektive durch den minimalen Kernquerschnitt ersetzt, um Sättigungserscheinungen im magnetischen Kern auszuschließen.

Diese Gleichung gilt nur, wenn die Wechselrichterausgangsspannung  $U_i$  symmetrisch ist, also für gleichlange Rechteckspannungspulse gleich großer positiver wie negativer Spannungsspitzenwerte. Wird ein variables Tastverhältnis bei gleichzeitiger Sicherstellung der Entmagnetisierung des Kerns zugelassen, ist unter Einbeziehung des Tastverhältnisses der Spannungsspitzenwert im Zähler von Gleichung 1.9 durch die effektive Spannung zu ersetzen ( $\hat{U}_i = \mathbf{n} \cdot U_{i,\text{eff}}$ ).

Nun können die Gleichungen 1.39 und 1.40 in Gleichung 1.38 eingesetzt werden. Man erhält die vom Tastverhältnis unabhängige Gleichung:

$$P_{\text{Übertragen}} = \frac{2}{\sqrt{\mathbf{n}_{0,\max}}} \cdot \hat{B} \cdot f \cdot A_{\min} \sqrt{\frac{P_N}{l_N \cdot \mathbf{r}_{\text{Draht}, T_{\max}} \left( \frac{\mathbf{g}_{N1}}{\mathbf{e}_{N1} \cdot A_N} + \frac{\mathbf{g}_{N2}}{\mathbf{e}_{N2} \cdot A_N} \right)}} \quad (1.41)$$

Diese Gleichung läßt sich noch vereinfachen, indem folgende Annahmen getroffen werden:

- 1) Die Veränderung des Leitungswiderstandes durch Skin- und Näherungseffekt wird in Primär- und Sekundärwicklung ( $\mathbf{g}_{N1}$  und  $\mathbf{g}_{N2}$ ) als annähernd gleich angenommen. Dies setzt eine Kammerwicklung oder ineinander geschachtelte Wicklungen voraus. Diese Veränderung wird direkt in  $\mathbf{r}_N$  zusammen mit dem spezifischen Drahtwiderstand bei maximal zulässiger Arbeitstemperatur  $\mathbf{r}_{\text{Draht}, T_{\max}}$  ausgedrückt.

2) Die Wickelraumfüllfaktoren  $\mathbf{e}_{N1}$  und  $\mathbf{e}_{N2}$  sind wie  $\mathbf{n}_{0,\max}$  hauptsächlich vom Schaltnetzteilprinzip abhängig und werden daher in einem vom Übertragertyp unabhängigen Leistungsfaktor  $K$  zusammengefaßt:

$$K = \sqrt{\frac{\mathbf{e}_{N1} \cdot \mathbf{e}_{N2}}{(\mathbf{e}_{N1} + \mathbf{e}_{N2}) \cdot \mathbf{n}_{0,\max}}} \quad (1.42)$$

Damit vereinfacht sich Gleichung 1.41 zu:

$$P_{\text{Übertragen}} = 2 \cdot \hat{B} \cdot K \cdot f \cdot A_{\min} \sqrt{\frac{P_N \cdot A_N}{l_N \cdot \mathbf{r}_N}} \quad (1.43)$$

Im weiteren Verlauf der Herleitung optimierter Auslegungsgleichungen ist die Ableitung der entstehenden Verlustwärme miteinzubeziehen. Hier ist zu unterscheiden, in welchem Maße die Wicklungsverluste durch den Kern abgeführt werden müssen. Die Unterscheidung wird anhand der unterschiedlichen Bauformen getroffen.

Kerntyp	Temperaturanstieg
Ringkerne, E-, ETD-, EC-, ER-, EFD-, U- und UI-Kerne	$\Delta \mathbf{J}_K = R_{th,K} \cdot P_K + R_{th,N} \cdot P_N$ (1.44a)
RM-, PM-, P-, EP- und Q-Kerne	$\Delta \mathbf{J}_K = R_{th,K} \cdot P_K + (R_{th,K} + R_{th,N}) \cdot P_N$ (1.44b)

Gleichung 1.44: Temperaturanstieg unterschiedlicher Kernformen

Die weitere Herleitung soll für die erstgenannten Kernformen durchgeführt werden. Die Gleichung für den Temperaturanstieg 1.44a wird nach  $P_N$  umgeformt und in Gleichung 1.43 eingesetzt. Danach wird Gleichung 1.15 mit der Kernmasse multipliziert und ebenfalls in die obige Gleichung eingesetzt. So erhält man einen Ausdruck, der die mit einem bestimmten Kern übertragbare Leistung angibt. Variabel sind dabei Frequenz, Induktionshub, Temperaturanstieg, Spannungsform und thermischer Widerstand:

$$P_{\text{Übertragen}} = \Delta B \cdot K \cdot f \frac{A_{\min} \sqrt{A_N}}{\sqrt{R_{th,N} \cdot l_N \cdot \mathbf{r}_N}} \sqrt{\Delta \mathbf{J} - R_{th,K} \cdot m_K \cdot p_0 \cdot \left(\frac{F}{F_0}\right)^X \cdot \left(\frac{f}{f_0}\right)^Y \cdot \left(\frac{\Delta B}{\Delta B_0}\right)^Z} \quad (1.45)$$

Aus  $\frac{\partial P_{\text{Übertragen}}}{\partial \Delta B} = 0$  folgt der optimale Induktionshub:

$$\Delta B_{\text{Opt}} = \Delta B_0 \cdot \sqrt[Z]{\frac{2}{Z+2} \frac{\Delta J_K}{R_{th,K} \cdot m_K \cdot p_0 \cdot \left(\frac{F}{F_0}\right)^X \cdot \left(\frac{f}{f_0}\right)^Y}} \quad (1.46)$$

In Gleichung 1.37 werden die Wicklungswiderstände gemäß Gleichung 1.39 ersetzt. Der Effektivwert des Stroms wird durch den Effektivwert der Stromdichte im Wickelraum ersetzt. Die Wicklungsverluste können gemäß Gleichung 1.44a durch den Temperaturanstieg und die Kernverluste ausgedrückt werden. Die Kernverluste werden entsprechend Gleichung 1.15 durch sekundäre Größen ausgedrückt. Es ergibt sich eine Beziehung zwischen der Stromdichte  $J$  und dem Induktionshub  $\Delta B$ , die einen bestimmten Temperaturanstieg im Kern bewirken:

$$J = \sqrt{\frac{\Delta J_K - R_{th,K} \cdot m_K \cdot p_0 \cdot \left(\frac{F}{F_0}\right)^X \cdot \left(\frac{f}{f_0}\right)^Y \cdot \left(\frac{\Delta B}{\Delta B_0}\right)^Z}{R_{th,N} \cdot r_N \cdot l_N \cdot A_N}} \quad (1.47)$$

Setzt man in diese Gleichung den Ausdruck für den optimalen Induktionshub ein, erhält man die optimale Stromdichte:

$$J_{\text{Opt}} = \sqrt{\frac{Z}{Z+2} \frac{\Delta J_K}{R_{th,N} \cdot r_N \cdot l_N \cdot A_N}} \quad (1.48)$$

Die Gleichung 1.45 vereinfacht sich nun zu:

$$P_{\text{Maximal}} = K \cdot f \cdot A_{\text{min}} \cdot A_N \cdot \Delta B_{\text{Opt}} \cdot J_{\text{Opt}} \quad (1.49)$$

Aus Gleichung 1.40 ergibt sich zur Dimensionierung der Primärwicklung:

$$N_1 = \frac{n_{\text{max}} \cdot U_{i,\text{min}}}{A_{\text{min}} \cdot \Delta B \cdot f} \quad (1.50)$$

Mit den Gleichungen 1.46, 1.48 und 1.50 läßt sich der Mittelfrequenztransformator optimieren. Es ist jedoch nicht einfach, die Exponenten  $X$ ,  $Y$



und  $Z$  zu erhalten. Entsprechende Daten liegen für VITROVAC<sup>®</sup> 6030F, Fix 350 vor [M29].

Nötigenfalls müssen die Optimierungsgleichungen ab den Gleichungen 1.43 und 1.44 geeignet hergeleitet werden. Die Ausdrücke für die Kernverlustleistung  $P_K$  sind dann entsprechend der obigen Darstellung einzusetzen und abzuleiten. Lediglich die Gleichung für die optimale Stromdichte bleibt dann erhalten.

### 1.7.6.2 Transformatorauslegung mittels Nomogrammen

Sofern ein Lösungsweg mittels geschlossener Gleichungen gangbar ist, ist dieser zu bevorzugen. Diese Berechnungsvorschriften wurden in Kapitel 1.7.6.1 nachvollzogen. Hierfür ist eine große Zahl von materialbeschreibenden Parametern einschließlich des thermischen Widerstands des Transformator-kerns notwendig. Diese Parameter sind insbesondere für die Metallbandkerne der Firma VAC Vakuumschmelze sowie die EFD-Kerne aus Ferriten der Firma Philips verfügbar.

Für die übrigen Ferritkerne verbleibt derzeit nur der in [M10] vorgestellte Lösungsweg. Dies gilt insbesondere für die Materialien der Firma Siemens. Die Lösung basiert auf Auslegungsdiagrammen, auch Nomogramme genannt. Ein Nomogramm beinhaltet Kurvenscharen in einem Koordinatensystem der übertragenen Leistung über dem Induktionshub, die experimentell für ein Kernmaterial bei einem bestimmten Kerntemperaturanstieg für eine festgelegte Schaltnetzteiltopologie ermittelt worden ist.

In [M10] ist eine große Zahl solcher Nomogramme veröffentlicht, mit deren Hilfe unter Zuhilfenahme von Interpolationsverfahren für Zwischenwerte eine Transformatorauslegung zu guten Ergebnissen führen kann.

Der Lösungsweg, der dem Schaltnetzteilentwickler angeboten wird, soll am Beispiel von Bild 1.17 verdeutlicht werden. Auf der Suche nach einem Ferritkern für Halb- oder Vollbrückenwandler, der eine bestimmte Leistung übertragen kann, sind sechs Arbeitsschritte im Nomogramm auszuführen:

- Zunächst ist eine waagrechte Linie bei der gewünschten Leistungsabgabe ( $P_{thr}$  in Bild 1.17) zu ziehen.
- Im zweiten Schritt wird der Schnittpunkt dieser Linie mit der gewünschten Schaltfrequenz bestimmt.
- Danach wird eine Parallele zur Frequenzachse ermittelt, die durch den Schnittpunkt verläuft, um die minimale Kernmasse ( $m_c$  in Bild 1.17) zu erhalten.

- Viertens wird unter den verfügbaren Magnetkernen der leichteste Kern gesucht, dessen Masse  $m_c$  größer oder gleich dem benötigten Wert ist.
- Fünftens wird wieder - ausgehend von der Masse des gefundenen Kerns - eine Parallele zur Frequenzachse gezogen.
- Der Schnittpunkt mit der Linie für die benötigte Leistung legt schließlich den optimalen Induktionshub (an der horizontalen Achse abzulesen) und die Schaltfrequenz fest.

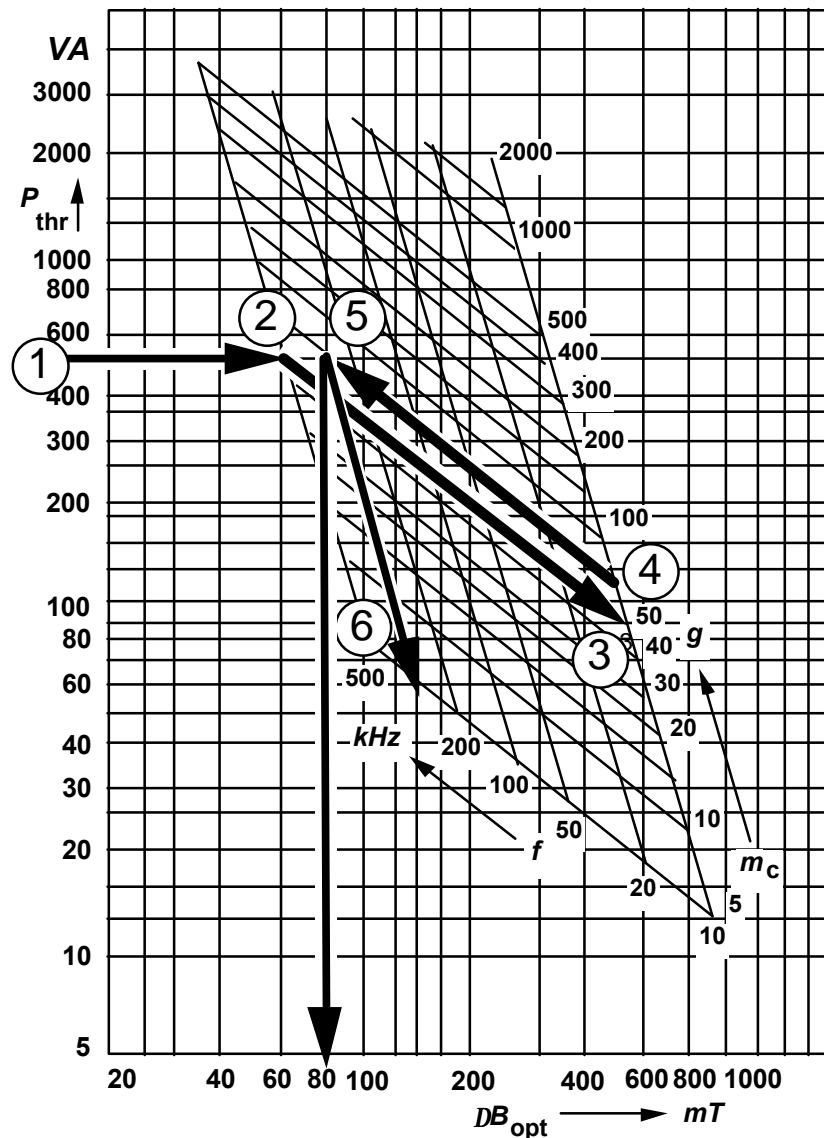


Bild 1.17: Nomogramm für Brückentopologien für einen Temperaturanstieg von  $DJ_K = 30$  K

Mit den gefundenen Werten kann gemäß Gleichung 1.50 die Primärwindungszahl festgelegt werden. [M10] beschreibt darüberhinaus das Vorgehen, wenn mehrere Sekundärkreise vorzusehen sind.

$$P_{thr,red} = c_1 c_2 P_{thr,max} \quad (1.52)$$

Die Korrekturfaktoren  $c_1$  und  $c_2$  vermindern die maximal übertragbare Leistung des Transformators. Dabei ist  $c_1$  vom Übersetzungsverhältnis,  $c_2$  von der Anzahl der Sekundärwicklungen abhängig. Dementsprechend ist zum Beispiel ein größerer Kern zu wählen, wenn mehrere Sekundärwicklungen benötigt werden.

## 1.8 Zusammenfassung

Schwerpunkt des 1. Kapitels war die Einführung in das Problemfeld Transformatorauslegung für Schaltnetzteile. Es konnte gezeigt werden, daß der Optimierungsvorgang von einer großen Zahl von Detailinformationen abhängig ist, die von den Herstellern von Magnetmaterialien nicht umfassend bereitgestellt werden. Statt dessen bieten die Hersteller dem Schaltnetzteilentwickler unterschiedliche Lösungswege an, die jeweils auf Näherungen und Annahmen, sowie dem Erfahrungswissen der Firmen basieren. Dabei muß oft auf Nomogramme zurückgegriffen werden, aus denen die Werte grafisch zu ermitteln sind. Im Bemühen um Automatisierung der Auslegung sind diese nur mühevoll durch Regressionsberechnungen in geschlossene Kurvenformen zu überführen. In jedem Fall stellt sich der Lösungsweg bei verschiedenen Herstellern unterschiedlich dar, was den Auslegungsvorgang wenig komfortabel und zeitraubend erscheinen läßt.

## **2. Möglichkeiten einer softwaretechnischen Umsetzung**

### **2.1 Bedarf einer wissensbasierten Teilautomatisierung**

Durch die Vielzahl möglicher Variationen, die im Bereich primärgetakteter Schaltnetzteile bei der Auslegung sowohl des Gesamtsystems Schaltnetzteil als auch bei der Auswahl des optimalen Leistungstransformators möglich sind, werden eine richtige Entscheidung für ein Schaltnetzteilprinzip sowie eine optimale Auslegung der Komponenten des Leistungstransformators sehr aufwendig und rechenzeitintensiv. In der Praxis bestimmt die langjährige Erfahrung von Spezialisten die Auswahl der Bauteile eines Schaltnetzteils. Der immer stärker werdende Kostendruck zwingt aber zu einer optimierten Auslegung dieser Komponenten. Überdimensionierte Sicherheitsreserven müssen wegen knapper Kostenkalkulation und erhöhter Anforderungen an Gewicht und Größe des Gesamtsystems entfallen. Die Dimensionierung per Hand erfordert Berechnungen der in Kapitel 1 hergeleiteten Auslegungsgleichungen.

Die Ergebnisse der Transformatoroptimierung können dazu führen, daß Vorgaben des Schaltungsprinzips nicht eingehalten werden. Dies macht es oftmals erforderlich, Entscheidungsbäume wiederholt zu durchlaufen. Berechnungen für die vielen möglichen Komponenten müssen mehrfach durchgeführt werden. Die Ergebnisse sind nach sich verändernden Kriterien auszuwerten, wenn beispielsweise die Vorgaben an das Gesamtsystem geändert werden mußten. Die Berechnung muß auch dann mehrmals mit unterschiedlichen Gleichungen durchgeführt werden, wenn Kerne für Transformatoren nur in bestimmten Abmessungen zur Verfügung stehen und die Optimierungsgleichungen teilweise kernform- und herstellerspezifisch sind.

Ein Programm, das den Entwickler an dieser Stelle unterstützen soll, muß einem hohen Maß an Aufgaben gerecht werden:

- Einerseits soll eine Vielzahl von Auslegungsalgorithmen bereitgehalten werden.
- Aus der Vielzahl der Kerngeometrien und Magnetmaterialien, in denen Kerne für Mittelfrequenztransformatoren verfügbar sind, ergibt sich der Bedarf der Pflege einer Produktdatenbank.
- Die große Zahl von Iterationsschritten bei einer kompletten Suche über die Grundgesamtheit der Kerne und Schaltnetzteilprinzipien würde eine Akzeptanz des Systems aufgrund unzumutbarer Antwortzeiten verhindern. Eine Stufe der Voreinschränkung dieser Grundgesamtheit ist not-

wendig. Der Benutzer sollte in dieser Phase teilnehmen können, da er hier sein Erfahrungswissen einbringen und die Berechnungsdauer verringern kann. Dazu muß das Programm in der Lage sein, über getroffene Entscheidungen Auskunft geben, sowie bei anstehenden Entscheidungen Empfehlungen aussprechen zu können.

- Die Mensch-Maschine-Kommunikation soll grafisch aufbereitet erfolgen, da beispielsweise Kennlinienverläufe, Schaltbilder und Explosionszeichnungen bewährte Arbeitsmittel für den entwickelnden Ingenieur sind.

Eine Software, die diesen Anforderungen genügen kann, wird sehr umfangreich. Eine komplette Neuprogrammierung von der grundlegenden Ablauforganisation über die Datenhaltung der Kernmaterialien und Kernformen, die Verwaltung der Auslegungsalgorithmen sowie der Zugriff auf diese bis hin zur Mensch-Maschine-Schnittstelle für den späteren Benutzer erweist sich als sehr umfangreich und zeitintensiv. Daher empfiehlt sich eine Analyse der Aufgaben, die im einzelnen zu bewältigen sind, sowie der softwaretechnischen Lösungskonzepte, die derzeit verfügbar sind. Eine Bewertung und Zuordnung dieser Konzepte soll deren jeweilige Einsetzbarkeit prüfen und schließlich die geeignetste Vorgehensweise erarbeiten.

Bei den zu bewältigenden Aufgaben handelt es sich um Arbeitsschritte, die ein Mensch sonst durchführen müßte, sowie Entscheidungen, die er fällen würde. Die Künstliche Intelligenz ist eine Disziplin, die sich mit der Nachbildung menschlicher Vorgehensweisen durch Automaten beschäftigt. Insbesondere die Umsetzbarkeit von Erfahrungs- oder heuristischem Wissen ist hier untersucht worden. Geeignete Werkzeuge stehen bereit. Daher sollen im folgenden neben Datenbankanwendungen vor allem KI-Produkte betrachtet werden.

### **2.1.1 Datenbanken**

Die Datenhaltung von Datenbankanwendungen erfolgt in Form von Tabellen. Jeder Tabelleneintrag ist ein Datensatz. Jeder Datensatz besteht aus Feldern. Datenbanken erweisen sich als besonders geeignet, große Mengen an gleichartigen Informationen verwalten zu können. Der Zugriff auf die gespeicherten Datenbestände erfolgt mittels Anfragen wie „Gebe alle Datensätze der Tabelle Kernformen aus, die im Feld Kernmaterial den Wert N27 und im Feld Kernformklasse den Eintrag E-Kern haben“. Änderungen der Daten erfolgen durch Lokalisierung einzelner Datensätze und Änderung des Wertes einzelner Felder [P7].

Diese Zugriffsoperationen werden in Datenbanksprachen wie SQL programmiert [P7]. Sie lassen sich aber auch einbetten in andere Programmiersprachen wie C, Cobol oder Pascal. Während SQL-Anweisungen von einer datenbankspezifischen Interpreteranwendung abgearbeitet werden, können eingebettete SQL-Anweisungen in beliebige kompilierte Anwendungen eingebracht werden. Dies ermöglicht eine flexiblere Programmgestaltung.

Vorteil der Verwendung einer Datenbank für die gestellte Aufgabe ist die Möglichkeit, die Materialdaten der unterschiedlichen Kerne zu erfassen. Andererseits zeigen die Erfahrungen aus Kapitel 1, daß die Größe einer derartigen Tabelle im Bereich von 100 Einträgen liegt. Für die Verwaltung der Vorgehensweisen ist keine besondere Unterstützung vorgesehen. Der notwendige Aufwand für die Beschaffung eines Datenbanksystems, die Einarbeitung und die eigentliche Softwareentwicklung ist deutlich höher als die erreichbare Verringerung des Programmieraufwands für die Gesamtanwendung. Eine einfachere Datenhaltung in Form einer Datei, die zur Programmlaufzeit in den Arbeitsspeicher des Rechners eingelesen wird, ist nicht zuletzt aufgrund des geringen Speicherbedarfs sowie des schnellen Zugriffs empfehlenswerter.

### **2.1.2 Objektorientierte Programmierung**

Bei der objektorientierten Programmierung steht nicht der Programmfluß durch Entscheidungsbäume im Vordergrund der Betrachtung. Vielmehr werden abstrakte Objekte definiert, die Merkmale aufweisen. Objekte, die durch gleichartige Merkmale beschrieben werden können und sich nur in den Merkmalswerten unterscheiden, können zu gemeinsamen Klassen von Objekten zusammengefaßt werden. Objekte mit Objektbeschreibungen, die eine Anzahl von Merkmalen gemein haben, sich aber in anderen unterscheiden, können durch Vererbung der Objektbeschreibungen von gemeinsamen Elternobjekten auseinander hervorgegangen sein.

Dieser beschriebene statische Zustand läßt die Erstellung einer ablauffähigen Anwendung nicht zu. Die Objekte besitzen Methoden, über die sie miteinander in Wechselwirkung treten können. Methoden können wie Merkmale vererbt werden. Darüberhinaus ist eine Initialmethode notwendig, um die Gesamtanwendung zu aktivieren. Der Programmfluß wird also fragmentiert in diesen Methoden verankert [P6, P11, P16].

Die Gesamtanwendung ließe sich realisieren, indem eine Klasse von Objekten die einzelnen Kerne repräsentieren würde. Pro Kern könnten die

Auslegungsalgorithmen als Methoden zugeordnet werden. Der Programmieraufwand könnte in Grenzen gehalten werden, indem alle Kerne mit gleichem Auslegungsalgorithmus zu Unterklassen zusammengefaßt würden und die nur die Gemeinsamkeiten aus der Elternklasse Kern erben. Andere Objekte müßten eingeführt werden, wie die Benutzerschnittstelle. Diese müßte den Sitzungsablauf aufgrund benutzerseitiger Eingaben und Reaktionen kontrollieren.

Diesem Konzept fehlt die Möglichkeit, auf der Suche nach dem geeignetsten Kern vorab gefallene Entscheidungen, die in den Methoden verankert sind, zurückzuverfolgen oder zu hinterfragen, warum und unter welchen Umständen sie getroffen worden waren. Der Benutzer einer Software, die Assistenz bei der Transformatorauslegung bieten will, ist durchaus interessiert, nachzuverfolgen, warum und an welchen Stellen des Entscheidungsprozesses beispielsweise die Grundgesamtheit der zur Verfügung stehenden Kerne eingeschränkt worden ist. Hierfür wäre eine geeignete Ablaufhistorie samt Auswertungskomponente noch zu realisieren.

### **2.1.3 Regelbasierte Systeme**

Produktionsregeln sind Wenn-Dann-Aussagen. Wenn der Bedingungs- teil einer Regel zutrifft, dann wird der Aktionsteil der Regel ausgeführt. Regeln werden in Regelmengen zusammengefaßt. Mit diesen Regeln werden Datenbestände kontrolliert verändert. Beispielsweise mögen zwei Variablen X und Y existieren. Auf diese sollen die Regeln einer Regelmenge angewandt werden. Eine dieser Regeln heiße: wenn  $X < 3$  und  $Y > 4$  ist, dann soll  $X = 5$  sein. Kommt diese Regel zur Bearbeitung, wenn gerade  $X = 2$  und  $Y = 10$  ist, dann wird die Variable X verändert und damit der Datenbestand im Ganzen. Eine andere Regel könnte jetzt ausführbar geworden sein. Die Reihenfolge, in der Regeln getestet und daraufhin gegebenenfalls ausgeführt werden, wird von einer Abarbeitungsstrategie bestimmt. Die Verarbeitungseinheit, die ein derartiges Regelwerk gemäß einer vorgegebenen Strategie verarbeitet, heißt Inferenzmaschine. Diese zeichnet auch die Abarbeitungsreihenfolge auf, sodaß der Benutzer den Fortschritt der Problemlösung rückverfolgen und hinterfragen kann.

Vorteilhaft ist, daß die Strategie und die Regeln einer Menge nicht notwendigerweise verknüpft sein müssen. Die Regeln können zusammenge- tragenes Erfahrungswissen beinhalten, das formuliert worden ist. Nicht be- nötigte Wenn-Dann-Aussagen werden nicht zur Ausführung kommen. Sie

erhöhen nur die Bearbeitungszeit einer Regelmenge um unnötige Überprüfungen der Bedingungssteile nicht benötigter Regeln [X1-X7].

Mit Regelmengen ließen sich vorteilhaft Einschränkungen der Grundgesamtheit der Schaltungsprinzipien und Kerne vornehmen. Die Auslegungsalgorithmen könnten in den Aktionsteilen bestimmter Regeln untergebracht werden. Andererseits ist in rein regelbasierten Systemen definitionsgemäß keine Aussage über die Möglichkeiten der Datenrepräsentation gemacht worden. Dies stellt einen Freiheitsgrad der Inferenzmaschine dar, der weiterer Klärung bedarf.

#### **2.1.4 Zwanggesteuerte Systeme**

Constraints formulieren Zwänge. Sie beschreiben Abhängigkeiten zwischen Größen und sind geeignet, deren Einhaltung zu überwachen. So kann die Einhaltung des Ohmschen Gesetzes bei einem Widerstand  $U = R I$  überwacht werden. Sind  $U$  und  $R$  bekannt und die Variable  $I$  ist uninitialisiert, läßt sich deren Wert durch den Constraint ermitteln. Ist  $I$  dagegen ein gültiger Wert, der nicht  $U / R$  ist, löst der Constraint einen Fehler aus, auf den geeignet reagiert werden kann. Constraints lassen sich zu Netzwerken zusammenbinden durch gemeinsamen Zugriff auf bestimmte Größen [X9].

Auch bei rein zwanggesteuerten Systemen ist keine Aussage über die Datenrepräsentation gemacht worden. Die Auslegungsalgorithmen für Schaltnetzteiltransformatoren ließen sich bei Constraintverletzungen aus den Fehlerbehandlungsroutinen heraus aufrufen. Vorentscheidungen, die zu einer Einschränkung der Grundgesamtheit führen, lassen sich aber nur aufwendig realisieren.

#### **2.1.5 Weitere Techniken der künstlichen Intelligenz**

Denkbar wären weiterhin die Verwendung von Fuzzy Control und neuronalen Netzen. Diese sind wie die regelbasierten und zwanggesteuerten Systeme im Rahmen der Erforschung künstlicher Intelligenz [X8] entstanden. Bei der Fuzzy Control werden Eingangsgrößen zunächst bewertet, danach klassifiziert und einem Regelwerk zugeführt [X10]. Die Aktionsteile dieser Regeln entscheiden über die Ausgestaltung der Ausgangsgrößen im Zusammenhang mit Bewertungsfunktionen. So lassen sich komplexe Regler aufbauen.

Interessant für die hier gestellte Aufgabe erscheint in diesem Zusammenhang die Fähigkeit, vage Aussagen zu verarbeiten, also beispielsweise die Zuordnung exakter Werte wie 4,3 mA zu Aussagen wie „ein geringer



Strom“. Man spricht hier von linguistischen Variablen. So lassen sich qualitative Aussagen in Regelwerken verarbeiten und anschließend exakt quantifizieren.

Künstliche neuronale Netze sind nach dem biologischen Modell der Verknüpfung menschlicher Nervenzellen erbaut. Sie sind in der Lage, analoge Eingangssignale zu verarbeiten mit dem Ziel, eine analoge Ausgangsgröße geeignet zu verändern. Sie können ebenfalls als intelligente Regler eingesetzt werden. Um die unterschiedlichen Neuronen zu programmieren, ist ein Lernverhalten vorgesehen. Die Abweichung der gewünschten Ausgangsgröße von der tatsächlichen wird als Fehler in das neuronale Netz zurückgespeist. Die Neuronen sind Summierglieder mit Bewertungsfunktionen der Eingangsgrößen. Die Vielzahl von Parametern wird in der Lernphase solange angepaßt, bis das gewünschte Ausgangsverhalten erreicht ist.

Die Fähigkeit der neuronalen Netze zu lernen wird im Rahmen der zu erstellenden Software so nicht benötigt. Die Einpflege neuer Kerndaten und Auslegungsalgorithmen erfolgt einmalig. Ein Lernen während diverser Sitzungen ist nicht vorgesehen.

### **2.1.6 Entscheidung zugunsten hybrider Expertensystems**

Die bisher kurz vorgestellten Softwarekonzepte haben jeweils Stärken und Schwächen hinsichtlich der zu bewältigenden Aufgabe gezeigt. Vorteilhaft wäre ein Werkzeug, das diese Softwarekonzepte so vereint, daß sich die jeweiligen Stärken ergänzen. Im Vorgriff auf das Kapitel 2.2 sei bemerkt, daß derartige Systeme existieren. Man spricht hier von hybriden Expertensystems. Mit ihnen lassen sich Expertensysteme, auch wissensbasierte Assistenzsysteme genannt, erstellen.

In Kapitel 1 ist gezeigt worden, daß die Auslegung von Mittelfrequenztransformatoren für primärgetaktete Schaltnetzteile ein umfangreiches Unterfangen darstellt, bei dem viele Einflußgrößen in wechselseitiger und nichtstetiger Abhängigkeit zueinander stehen. Ein wissensbasiertes System kann den Zeitaufwand für die Auslegung stark verringern, indem es das notwendige Fachwissen bereitstellt und Hilfestellung bei der Entscheidungsfindung anbietet. Es kann auch einen routinierten Schaltnetzteilentwickler unterstützen und die Entwicklungszeit reduzieren. Eine derartige Aufgabe ist geeignet, in Form eines Expertensystems behandelt zu werden. Die große Bedeutung des Kontrollwissens für die Auslegung von Kern und Wicklung eines Schaltnetzteiltransformators führt weg von einer linearen Problemlö-

sung hin zu einem offenen System, in das erfahrene Mitarbeiter ihre Kenntnisse der bisherigen Auslegungen einbringen können.

Expertensysteme für Anwendungen in Bereichen der Leistungselektronik wie der Entwicklung, Fertigung und Produktauswahl sind in aktuellen Veröffentlichungen [T2-T4] vorgestellt worden. Darin werden Expertensysteme beschrieben, die ein breites Spektrum an Fachwissen abdecken wollen, ohne dabei auch die Lösung tiefgreifender technischer Probleme zu ermöglichen. Unter Berücksichtigung der derzeit verfügbaren Rechenleistung und des Arbeitsspeichers konventioneller Workstations ist es erforderlich, daß sich wissensbasierte Anwendungen auf Wissensgebiete beschränken, die scharf umrissen sind, dafür aber tief gründen. Expertensysteme für derartige Wissensgebiete können mit noch akzeptablen Antwortzeiten realisiert werden.

## **2.2 Hybride Expertensystemschaalen**

Ein Mensch wird in der Regel dann als Experte auf einem Fachgebiet bezeichnet, wenn er sich durch eine geeignete Ausbildung oder seine langjährige Tätigkeit spezielles Wissen und Problemlösungsfähigkeiten angeeignet hat, die ihm das Bewältigen von Aufgabenstellungen ermöglichen, die von einem Nicht-Experten nicht oder nicht so effizient zu bewältigen wären. In diesem Zusammenhang wird auch von Sachverständigen gesprochen. Läßt sich solches Wissen nicht in klare Aussagen fassen, sondern liegen ausschließlich Erfahrungswerte vor, die sich mit den Jahren zu einer immer wieder gleichbleibenden Methodik entwickelt haben, spricht man von heuristischem Wissen oder von Heuristiken.

Bei Expertensystemen handelt es sich um Computerprogramme, die Wissensrepräsentations- und Wissensverarbeitungsmechanismen kennen, die sie in die Lage versetzen können, auf einem speziellen Wissensgebiet die Kompetenz eines menschlichen Experten nachzubilden. Sie können als Beratungs- und Lernsysteme oder als Diagnosesysteme eingesetzt werden. Sie können Auslegungsprobleme bewältigen und komplexe technische Prozesse überwachen und steuern. Außer bei der letztgenannten Funktion muß das Programm immer in der Lage sein, Anfragen eines Benutzers entgegenzunehmen, zu präzisieren und in vollständige Problemstellungen umzuformulieren. Es muß verständliche Antworten generieren und Hilfestellung während der gesamten Sitzung anbieten können. Diese Anforderungen lassen sich mit herkömmlichen Datenverarbeitungsprogrammen nicht oder nur mit einem sehr hohen Aufwand realisieren [X1].

Allein der Begriff Expertensystem verleitet leicht zu der Annahme, das Wissen eines Fachmanns nach geglückter Eingabe ständig in einer Weise verfügbar zu haben, daß auf diesen Menschen nötigenfalls ganz verzichtet werden kann. Praktisch stößt man im Umgang mit derartigen Systemen bald an Grenzen, die dann die überhöhten Erwartungen schnell enttäuschen. Die Verwendung derartiger Systeme führt nicht zwangsläufig zum gewünschten Erfolg [X2].

Mit den heute zur Verfügung stehenden Expertensystemshalen - also Programmen zur Erstellung und dem Betrieb von Expertensystemen - ergeben sich Zwänge im Bereich der Inferenzmaschine, also der Problemlösungskomponente, des Speicherbedarfs und der vom Benutzer noch akzeptierbaren Antwortzeiten. Das hat zur Folge, daß eine scharfe Eingrenzung des verfügbaren Wissens vorgenommen werden muß, um die notwendige Wissenstiefe zu erreichen. Nur wenn sich das Endprodukt als eine kompetente Wissensquelle herausstellt - und sei es auch nur auf einem schmal umrissenen Fachgebiet - hat das Expertensystem eine Chance, akzeptiert zu werden. Nur dann rentiert sich der hohe Aufwand, der im Vorfeld in die Einarbeitung in die verwendete Schale, die Wissensakquisition, Wissensstrukturierung und abschließende Umsetzung in die verfügbaren Wissensrepräsentationsmechanismen erbracht werden muß.

### **2.2.1 Komponenten eines Expertensystems**

Die ersten Expertensysteme sind als Einzelprogramme realisiert worden. Ein sehr bekanntes Beispiel hierfür ist das an der Stanford University entwickelte Expertensystem MYCIN zur Diagnose und Therapie von bakteriellen Infektionskrankheiten des Blutes und der Meningitis [X2-X4]. Um nicht für jedes andere Problem, das wissensbasiert gelöst werden soll, vollständig neu entwickeln zu müssen, entstand daraus die Expertensystemshale EMYCIN (oder Empty MYCIN). Zu diesem Zweck war das medizinische Fachwissen aus MYCIN entfernt worden, sodaß nur die Problemlösungsfähigkeiten verblieben. Die Trennung von Schale - also Ablaufumgebung oder Programm - und Wissensbasis - beziehungsweise einzulesender Datei - leistete der Verbreitung von Expertensystemen Vorschub.

Den Kern eines jeden Expertensystems bildet die Inferenzmaschine, auch Problemlösungs- oder Schlußfolgerungskomponente genannt. Sie ist insbesondere in der Lage, unterschiedliche Wissensaussagen zu einer Kette von Schlußfolgerungen zu verbinden, sowie Schlüsse aus den Wissensaussagen zu ziehen, die in Form von bereits bekannten Fakten und Regeln für

das weitere Vorgehen vorliegen. Vorhandenes Wissen kann verknüpft werden, um hieraus neues Wissen abzuleiten. Die Wissensaussagen stehen in der Wissensbasis zur Verfügung.

In Bild 2.1 wird das Zusammenspiel dieser zentralen Elemente eines Expertensystems mit den weiteren Komponenten aufgezeigt, die im folgenden beschrieben werden. Der graue Bereich verdeutlicht das Expertensystem mit seinen unterschiedlichen Schnittstellen.

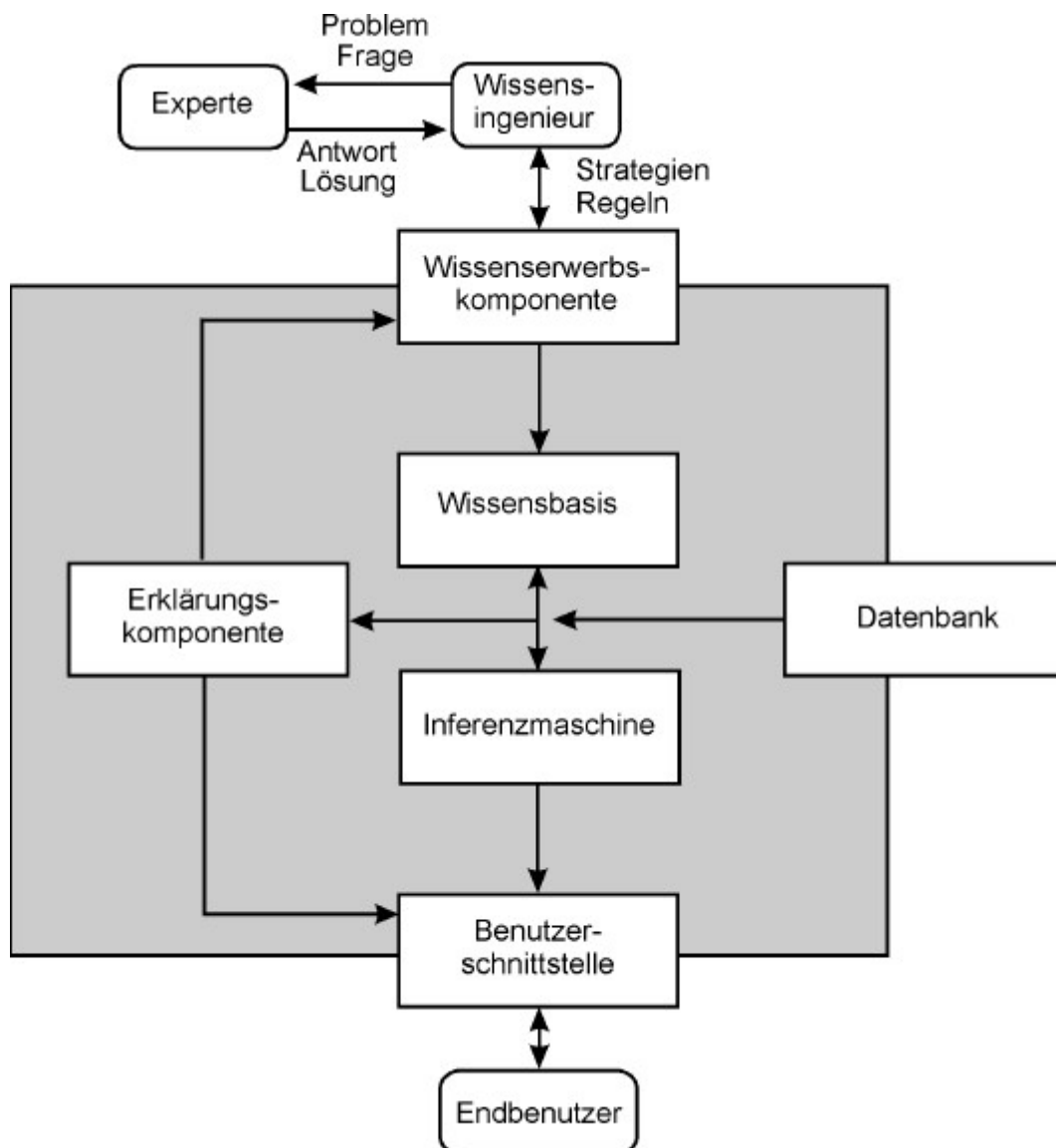


Bild 2.1: Komponenten eines Expertensystems

Der Inferenzmaschine kann darüberhinaus ein Datenbankanschluß zur Verfügung stehen, um ihr den Zugriff auf große Datenmengen zu ermöglichen.

chen. Bei den meisten Schalen stehen der Inferenzmaschine zusätzlich die Fähigkeit zur Abarbeitung und Auswertung der Ergebnisse von Algorithmen, das Aktivieren externer Programme inklusive Datenaustausch, sowie grafische Möglichkeiten zur Kommunikation mit dem Benutzer zur Verfügung.

Die Schale stellt mit ihren Möglichkeiten der Wissensrepräsentation die Rahmenbedingungen für die Wissenseingabe, da die Wissensbasis nur mit solchen Wissenskonstrukten gefüllt werden kann, die die Inferenzmaschine und deren periphere Komponenten bearbeiten können.

Um das Fachwissen in die Wissensbasis zu integrieren, bieten viele Schalen eine Wissenserwerbskomponente an. Die Eingabe einer Vielzahl von Daten, Regeln und Zwängen erfolgt zumeist in dafür ausgelegten Editoren. Diese unterstützen den Fachmann, der sein Wissen in das Expertensystem eingeben will. Falls von der Schale keine ausreichende Unterstützung für den Fachexperten bereitgestellt wird oder falls er nicht mit KI-Technologien vertraut ist, muß dieser einen Wissensingenieur bemühen. Der ist mit der Art und Weise der Wissenseingabe vertraut und kann - im Gespräch mit dem Fachexperten - die Wissensbasis erzeugen. Die Editoren der Wissenserwerbskomponente interpretieren die in formaler Sprache eingegebenen Wissensstrukturen, um sie geeignet in die Wissensbasis einbringen zu können.

Eine Benutzerschnittstelle erleichtert bei dialogorientierten Expertensystemen die Kommunikation zwischen Mensch und Inferenzmaschine in hohem Maße. Ihrem Aufbau kommt eine besondere Bedeutung zu, da sie das System repräsentiert und seine Akzeptanz stark mitbestimmt.

Ein weiterer wichtiger Teilbereich eines Expertensystems ist die Erklärungs- und Hilfefunktion. Sie kann dem Benutzer bereits gezogene Schlußfolgerungen des Expertensystems verdeutlichen oder ihn beim weiteren Vorgehen im Verlauf einer Sitzung beraten. Mit ihrer Hilfe kann der vom System gewählte Lösungsweg nachvollzogen werden. Dies macht die Vorgehensweise des Expertensystems für den Entwickler wie für den Benutzer transparent.

Ein Expertensystem muß nicht zwingend alle genannten Komponenten enthalten. Beispielsweise können eingebettete Systeme, die unmittelbar in einen technischen Prozeß eingebunden sind, direkt mit diesem kommunizieren. Ihre Meß- und Zustandsdaten bekommen solche Systeme über einen direkten Zugriff auf den Prozeß. Ein solches System kann ohne weiteres auf eine Erklärungskomponente verzichten. Eine benutzerfreundliche Kommunikationsschnittstelle kann ebenso entfallen [S4, X1].

Expertensysteme sind demnach grundsätzlich unterteilbar in

- dialogorientierte Beratungssysteme, hier kommen der grafischen Endbenutzerschnittstelle sowie der Erklärungskomponente besondere Bedeutung zu, und
- prozeßorientierte Systeme zur Steuerung, Regelung und Überwachung in Echtzeit. Hier steht neben der Fähigkeit, Erkenntnisse geeignet zu protokollieren die Schnelle der Reaktion im Vordergrund.

Nicht alle Inferenzmaschinen basieren ausschließlich auf der Abarbeitung von Wenn-Dann-Aussagen. Es gibt beispielsweise auch Werkzeuge, die mittels Constraintnetzen rein zwangsbasiert arbeiten [X9, S2, S3]. Systeme, die mehrere unterschiedliche Wissensrepräsentationsformalismen anbieten, welche voneinander wechselseitig Gebrauch machen können, werden als **hybride Expertensystemschaalen** bezeichnet [X7, X8]. Damit kann Wissen immer in der geeignetsten Weise eingegeben werden, was die Größe der Wissensbasis verringert und die Wissensverarbeitung begünstigt.

### 2.2.2 Arten von Wissen

Bevor die Wissens eingabe in eine hybride Schale Erfolg haben kann, muß eine gründliche Recherche des umzusetzenden Fachwissens erfolgen. Danach muß dieses Wissen strukturiert und - entsprechend seiner Inhalte - den verschiedenen Arten von Wissen zugeordnet werden [X1]. Dabei handelt es sich um:

- **Wissen über Objekte:** Wissen über die Beschaffenheit oder über die Eigenschaften eines Objekts. Dies kann zum Beispiel das Wissen über die technischen Daten eines Motors sein (Motor XYZ,  $U_R = 380V$ ,  $I_R = 10A$ , ...).
- **Wissen über Beziehungen:** Hierbei handelt es sich um vergleichendes Wissen zwischen Objekten. Zum Beispiel die Aussage, daß Synchronmaschinen immer mit der elektrischen Frequenz der speisenden Spannung synchron drehen, im Gegensatz zu Asynchronmaschinen, die eine um den Schlupf verringerte Drehzahl besitzen.
- **Wissen über Vorgehensweisen:** Der Experte weiß, in welcher Reihenfolge er das Problem angehen muß, um zu einer schnellen Lösung zu kommen. Dies kann bedeuten, daß zuerst die Bemessungsspannung eines Motors mit der Netzspannung verglichen werden muß, bevor andere Daten betrachtet werden dürfen.
- **Wissen über Vorschriften:** Bei einer Problemlösung müssen verschiedene Gesetze und Grenzwerte eingehalten werden. Das Wissen über diese

Zwänge bildet das Vorschriftenwissen. Beispielsweise ist der Betrieb einer permanent erregten Synchronmaschine im Feldschwäcbereich ohne Verwendung eines exakten Lagegebers nur eingeschränkt oder gar nicht möglich.

## 2.2.3 Wissensrepräsentationsformen hybrider Expertensystemschalen

### 2.2.3.1 Objektorientierung

Ein semantisches Netz, wie es in Bild 2.2 dargestellt wird, ist ein grafisches Netzwerk von Knoten, die durch gerichtete, benannte Kanten miteinander verbunden sind [X1, X2, X8]. Die Knoten enthalten die Fakten, während die Kanten beliebige Relationen beschreiben. Durch die Verwendung von Beziehungen wie "ist ein" oder "hat die Eigenschaft" lassen sich semantische Netze hierarchisch strukturieren. Dies bedeutet, daß Oberklassen ihre Eigenschaften an Unterklassen - beziehungsweise an Objekte - vererben.

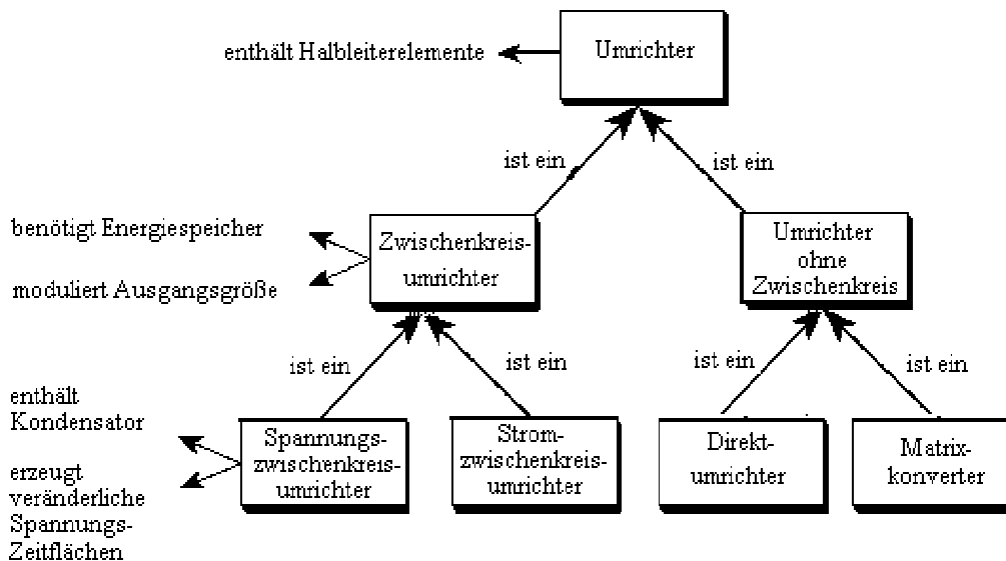


Bild 2.2: Aufbau eines semantischen Netzes

Im Beispiel aus Bild 2.2 existiert eine Oberklasse Umrichter, dem die Eigenschaft zugeordnet wird, Halbleiterventile zu besitzen. Diese Eigenschaft vererbt er an die Unterklassen Zwischenkreisumrichter und Umrichter-ohne-Zwischenkreis. Ein Zwischenkreisumrichter wiederum benötigt einen Energiespeicher und moduliert seine Ausgangsgröße. Die Unterklasse

Spannungszwischenkreisumrichter bekommt diese Eigenschaft nun übertragen (vererbt) und es werden weitere zugeordnet.

Mit Hilfe des semantischen Netzes kann folgende Aussage über einen Spannungszwischenkreisumrichter getroffen werden: ein Spannungszwischenkreisumrichter enthält als Energiespeicher einen Kondensator. Die veränderlichen Spannungszeitflächen werden über die Schaltzustände von Halbleiterventilen moduliert.

Semantische Netze bilden die konzeptionelle Grundlage für die Wissensdarstellung mit Hilfe einer objektorientierten Programmierweise. Bei objektorientierten Programmiersprachen stehen nicht Prozeduren im Vordergrund, die ungehindert auf beliebige Daten zugreifen können, sondern Objekte. Dabei handelt es sich um Datenstrukturen, deren einzelne Einträge, Slots, Attribute oder Merkmale genannt, die Eigenschaften dieses Objekts beschreiben. Der Zugriff auf diese Slots geschieht gemäß Kapitel 2.1.2 über Methoden, auch Behaviors oder Verhaltensweisen genannt. Sie sind den Objekten zugeordnet und können nur Merkmale des eigenen Objekts beeinflussen. Das heißt, der Zugriff auf Objektdaten geschieht immer nur über das betroffene Objekt selbst. Man spricht hier von Kapselung.

Ein Objekt ist wiederum eine individuelle Ausprägung einer Vielzahl gleichartiger Objekte, die einer gemeinsamen Klasse angehören. Dieser Programmierstil ist als eine Erweiterung des modularen Programmierstils entstanden. Er gestattet, selbst sehr große Programme noch handhaben zu können. Interessant wird er vor allem dadurch, daß durch Vererbungsmechanismen aus allgemeinen Objekten komplexere aufgebaut werden können. Beispielsweise könnte eine Schleifring-Asynchronmaschine aus der Rotor- und der Statorwicklung zusammengesetzt sein, die wiederum beide aus der Klasse Drehfeldwicklung stammen könnten. Neue Klassen werden also dadurch gebildet, daß Merkmale aus anderen Klassen ererbt werden und gegebenenfalls neue Merkmale hinzukommen. Auch die Behaviors werden vererbt, können aber durch individuelle Varianten ersetzt werden. Im Zusammenhang mit Expertensystemen werden Klassen häufig als Frames oder Rahmen und Objekte als Instanzen oder Ausprägungen bezeichnet.

Ein Frame entspricht in etwa einem Knoten innerhalb eines Semantischen Netzes. Im Gegensatz zu einem Semantischen Netz sind jedoch keine frei definierbaren Objektrelationen möglich. Durch Vererbung besteht bei Frames somit die Möglichkeit des hierarchischen Aufbaus hochspezialisierter Frames, beginnend bei einer allgemeinen Klasse (Super-Frame) bis hin zu immer spezielleren Klassen, den Sub-Frames.



In Bild 2.3 wird der Zusammenhang zwischen einem Frame und seinen Instanzen verdeutlicht. Der Frame Motor enthält die einzelnen Merkmale Name, Spannung, Strom, und so weiter. In den Instanzen des Frames Motor werden nun die speziellen Werte bestimmter Motoren eingetragen [X1, X2, X8, P5, P6, P15, P16]. Der Merkmalswert „-“ des Frames Motor in Bild 2.3 bedeutet, daß beim Erzeugen von Instanzen keine bestimmten Voreinstellungswerte gesetzt werden sollen. Es wäre aber ein spezialisierter Frame Drehstrommotor denkbar, der von dem Frame Motor abgeleitet sein könnte und bei dem ein zusätzliches Merkmal „Anzahl Phasen“ bereits mit 3 vorgelegt wäre, so daß dieser Wert nicht bei jeder Instantiierung zu setzen wäre.

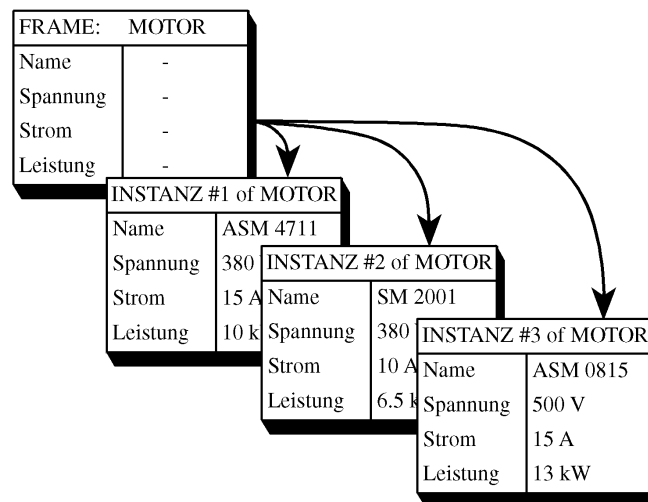
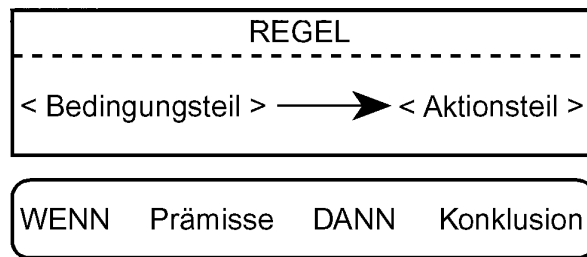


Bild 2.3: Frames und Instanzen

### 2.2.3.2 Produktionsregeln

Produktionsregeln, verkürzt als Regeln bezeichnet, sind die von den meisten Expertensystemen verwendete Form der Wissenspräsentation, da das in ihnen implementierte Wissen leicht zu lesen und zu pflegen ist. Sie bilden eine weitgehend kontextunabhängige Wissenseinheit, die daher auch isoliert betrachtet und verstanden werden kann. Hieraus ergibt sich eine einfache Modifizierbarkeit der Wissensbasis. Bild 2.4 verdeutlicht den Aufbau einer Regel, die im wesentlichen aus einem Bedingungsteil, der Prämisse, und einem Aktionsteil, der Konklusion, besteht.



WENN der Bedingungsteil (Prämisse) erfüllt ist,  
DANN wird der Aktionsteil (Konklusion) abgearbeitet.

Bild 2.4: Aufbau einer Regel

Durch das Anwenden von Regeln auf bereits bekannte Fakten können neue Fakten gewonnen werden.

Bei leistungsfähigen Expertensystemen wird die Gesamtzahl der eingebrachten Regeln schnell groß. Um die Suche nach geeigneten Regeln zu beschleunigen, können kontextbezogenen Regeln zu Regelmengen zusammengefaßt werden (Bild 2.5). Dieses Vorgehen empfiehlt sich, sobald ein Expertensystem insgesamt 100 oder mehr Regeln umfaßt, von denen für ein Teilproblem nur wenige Regeln benötigt werden, da auch die nicht zu diesem Kontext gehörenden Regeln immer wieder geprüft werden [X1, X2, X8].

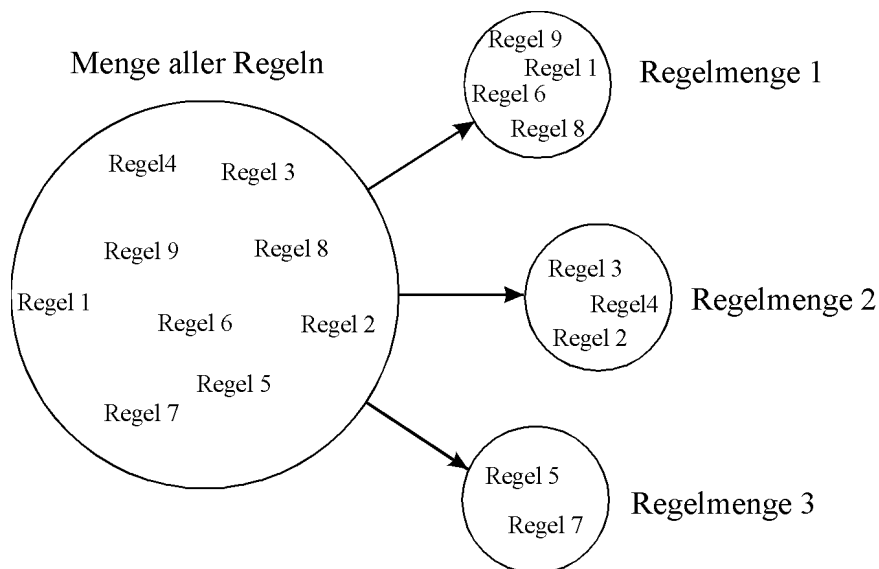
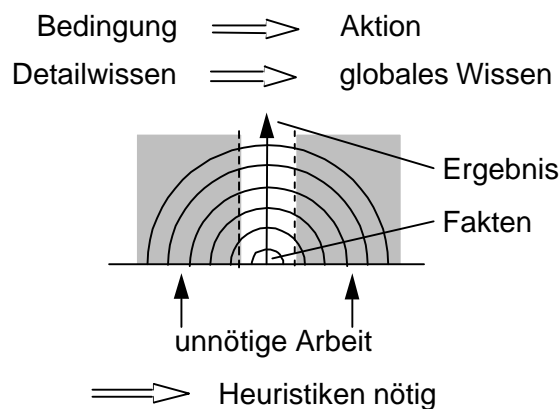


Bild 2.5: Aufteilung der Regeln einer Wissensbasis in Regelmengen

Es existieren drei grundsätzlich unterschiedliche Abarbeitungsstrategien für Regeln. Dies sind die Vorwärts- und Rückwärtsverkettung, sowie die sequentielle Abarbeitung:

- Bei der **sequentiellen Abarbeitung** werden den Regeln einer Regelmengen Prioritäten oder Vorränge zugeordnet. Die Regeln werden gemäß ihrer Prioritäten abgearbeitet. Bei mehreren Regeln gleicher Priorität werden diese Regeln gemäß der Reihenfolge ihres Auftretens abgearbeitet. Bei Verzicht auf den Mechanismus der Regelpriorisierung entscheidet die Platzierung der Regel in der Regelmengen-Liste über ihre Priorität. Die erste Regel hat automatisch die höchste Priorität.
- Bei der **Vorwärtsverkettung** werden zunächst alle anwendbaren Regeln der Regelmengen durch Mustervergleich des Bedingungssteils (pattern-matching) ermittelt. Anschließend wird eine Regel ausgewählt und zur Ausführung gebracht. Bei der Auswahl der Regeln wird zwischen Tiefensuche (Depth-first) und Breitensuche (Breadth-first) unterschieden. Vorwärtsverkettete Systeme werden auch als datengesteuerte Systeme bezeichnet [X1, X6]. Vorwärtsverkettete Regelmengen werden eingesetzt, wenn Teilergebnisse zu ermitteln sind und der Lösungsraum sehr groß ist. Heuristisches Vorgehen ist notwendig, um den Grad an nicht benötigtem Wissenserwerb in Grenzen zu halten. Dies verdeutlicht die eingegraute Fläche in Bild 2.6.

● Vorwärtsverkettung



● Rückwärtsverkettung

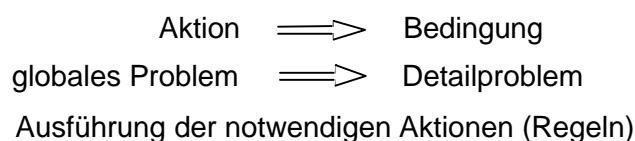


Bild 2.6: Strategien für die Verkettung von Regeln

- Im Gegensatz zu den vorwärtsverketteten Regelmengen geht die **Rückwärtsverkettung** von der Zielsetzung der Regeln, also der Konklusion, aus. Zunächst werden die Aktionsteile aller Regeln unter der Annahme geprüft, daß, wenn diese Regel zur Anwendung käme, das Ausführen des Aktionsteils zum gewünschten Teilziel führen würde. Trifft dies zu, werden die Voraussetzungen ermittelt, die den Bedingungsteil dieser Regel erfüllen würden. Es werden wieder die Aktionsteile aller Regeln durchsucht, um die Regeln zu finden, die die Erfüllbarkeit der Prämisse der aktuellen Ausgangsregel erreichen würden. Der Vorgang wird, beginnend bei dem tatsächlichen Zustand des Faktenwissens bis hin zur Lösung des anfänglich gesteckten Teilziels, solange wiederholt, bis die Reihenfolge der Regelarbeitung komplett festliegt. Falls dies nicht gelingen kann, wird abgebrochen. Rückwärtsverkettete Regelwerke werden auch als zielgesteuert bezeichnet (Bild 2.6).

### 2.2.3.3 Restriktionsüberwachung über Constraintnetze

Constraints, beziehungsweise Zwänge, bieten die Möglichkeit, Abhängigkeiten zwischen verschiedenen Merkmalen einzelner Instanzen zu beschreiben. Es besteht die Möglichkeit, Ausprägungen eines oder Kombinationen mehrerer Objektattribute einzuschränken. Constraints bilden eine Art übergeordnetes Wissen, auch Metawissen oder Wissen zweiter Art genannt. Mit Hilfe der Abhängigkeitsbeschreibungen eines Constraints können aus bekannten Werten eines bestimmten Constraints die unbekannt Werte ermittelt werden. Diese Vorgehensweise heißt Propagierung [X1, X2, X8, X9]. Es können aber auch gemäß Kapitel 2.1.4 Verletzungen dieser Abhängigkeiten überwacht werden. Entsprechende Sicherheitsabfragen beim Beschreiben von Slots erübrigen sich somit.

Mittels Bild 2.7 kann die Funktionsweise eines Constraints anhand des Ohmschen Gesetzes betrachtet werden. Es wird aus den Eingangswerten  $R$  und  $I$  die Ausgangsgröße  $U$  berechnet. Wird der Wert für die Spannung  $U$  in einer Weise geändert, daß die Beziehung  $U = R * I$  nicht mehr erfüllt ist, erfolgt eine Reaktion auf diesen Constraint-Fehler.

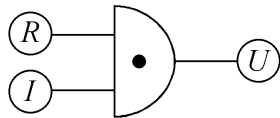


Bild 2.7: Ein einfacher Constraint

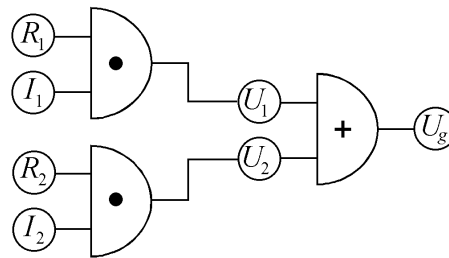


Bild 2.8: Ein Constraintnetz

In Bild 2.8 wird die Möglichkeit dargestellt, mehrere Constraints zu einem Netz zu verknüpfen. Ändert sich bei konstanter Ausgangsgröße eine der vier Eingangsgrößen, wird zunächst aufgrund der Constraintverletzung eine Inkonsistenz festgestellt. Eine geeignete Reaktion darauf ist in der Wissensbasis vorzusehen. Beispielsweise könnte die den Constraint verletzende letzte Aktion zurückgenommen werden. Waren bis dahin aber noch nicht alle Werte bekannt, wird das Constraint propagiert und die fehlende Ausgangsgröße ermittelt.

#### 2.2.3.4 Weitere Wissensrepräsentationsformen

Es gibt keine Festlegung darüber, welche der unterschiedlichen Wissensrepräsentationsmechanismen in einer hybriden Expertensystemschaale mindestens verfügbar sein sollten. Sicherlich sind die bisher genannten Bereiche die am häufigsten implementierten.

Es gibt aber auch Schalen, die weitere Mechanismen bereitstellen, wie Mittel der probabilistischen Wissensrepräsentation [X2]. Schlußfolgerungen werden hinsichtlich ihrer Glaubwürdigkeit bewertet. Beispielsweise können Aussagen, die nur aus Faktenwissen der ursprünglichen Wissensbasis abgeleitet worden sind, mit maximaler Glaubwürdigkeit, die Antworten auf Rückfragen beim Benutzer dagegen mit einer gewissen Unsicherheit versehen werden. Die weiteren Schlüsse, die das System zieht, werden dadurch immer unsicherer. Unterschiedliche Ergebnisse können so auch aufgrund ihrer Restsicherheit, ausgedrückt über den Certainty Factor, miteinander verglichen werden.

Aus dieser Möglichkeit, nicht nur kategorische Schlüsse zu ziehen, also nur die binären Wahrheitswerte wahr und falsch zuzulassen, kann eine Inferenzmaschine so erweitert werden, daß sie in der Lage ist, nicht-monoton zu schließen. Dabei wird unter Rückgriff auf die Historie des Schlußfolgerungsprozesses eine Wiederaufnahme an einem früheren Punkt möglich.

Beim temporalen Schließen kann dieser Vorgang zu bestimmten Zeitpunkten wiederangestoßen werden [X2].

Die Prädikatenlogik hat sich für deduktive Aufgaben bewährt. Sie stellt Methoden zur Verfügung, aus einer Menge von Einzelerfahrungen aussagenlogische Formeln abzuleiten und damit Prädikate so zu bilden, daß sich die Wissensbasis erweitern läßt. Man spricht von Wissenserwerb. Damit lassen sich beispielsweise aus einer großen Menge heuristisch erworbener Einzelaussagen allgemeingültige Regeln ermitteln [X2, P17, P18, S2, S3].

Für die hier gestellte Aufgabe der Transformatorauslegung werden derartige Wissensrepräsentationsformen aber weniger benötigt. Die Entscheidungen des Benutzers sind als kategorisch anzunehmen und nicht zeitlich abhängig. Zur Sitzungslaufzeit wird kein grundsätzlich neues Wissen erworben.

### **2.3 Auswahl einer geeigneten hybriden Expertensystemscha**

In Kapitel 2.1 sind die Anforderungen an das benötigte Softwareumfeld für die Erstellung eines Assistenzsystems zur Auslegung von Mittelfrequenztransformatoren für primärgetaktete Schaltnetzteile formuliert worden. In Kapitel 2.2 wurde erarbeitet, daß hybride Expertensystemschaen hierfür die besten Voraussetzungen bieten.

Es existieren heute weit über 100 verschiedene Expertensystemschaen unterschiedlicher Leistungs- und Preisklassen [S1, S4, X2-X6]. Es gibt hybride und rein regelbasierte Systeme mit Benutzerschnittstellen für Ein- und Ausgabedatenströme, Volltextbildschirme und Mehrfenstertechnik. Sie können unterschiedlich große Wissensbasen verarbeiten und sind auf verschiedenen Betriebssystemen lauffähig. Demgemäß sind Anschaffungskosten und Einarbeitungsaufwand auch sehr verschieden. Diese Werkzeuge sind unterschiedlich weit ausgereift, je nach dem Aufwand, der in die Versionspflege gesteckt wird. Dieser ist meist von der Zahl der Lizenznehmer und deren Wünschen auf Weiterentwicklung abhängig.

Andererseits ist gerade Anfang der neunziger Jahre eine Welle der Ernüchterung und damit ein Interesseverlust an wissensbasierten Lösungen festzustellen gewesen. Die zu hoch geschraubten Hoffnungen aus den achtziger Jahren in die Leistungsfähigkeit von Expertensystemen sind mit deren Inbetriebnahme enttäuscht worden. Die Entscheidungsträger in den Unternehmen wandten sich von dieser Technologie ab. Als Folge davon mußte ein großer Teil von Produkten vom Markt verschwinden.

Unter Berücksichtigung der genannten Aspekte ergibt sich folgendes Bild:

- Eine komplette Neuerstellung einer umfangreichen Assistenzsoftware ohne Rückgriff auf Entwicklungswerkzeuge macht wenig Sinn.
- Da im allgemeinen die Personalkosten für die Softwareentwicklung höher sind als die Kosten für die Beschaffung eines Entwicklungswerkzeugs und die dafür notwendige Einarbeitung, empfiehlt sich die Auswahl einer Softwarebasis, die die benötigten Leistungsmerkmale aufweist und verspricht, über Jahre hinaus weitergepflegt zu werden.
- Wenn die Schnittmenge dieser Auswahlkriterien zu klein ist, empfiehlt sich schließlich die Weiterentwicklung einer vorhandenen Softwarebasis mit teilweise unzureichenden Leistungsmerkmalen. Im Rahmen vorgegebener Randbedingungen muß eine Weiterentwicklung möglich sein. Dies schließt Betrachtungen des zu erbringenden Nachbesserungsaufwands ein.

Im folgenden wird daher ein Softwaresicherungskonzept vorgestellt, das es ermöglicht, eine leistungsfähige Softwarebasis für die hier zu lösende Aufgabe bereitzustellen. Durch konsequente Weiterentwicklung eines vorhandenen Werkzeugs, dessen Quelltexte frei von Auflagen verfügbar sind, auf die Bedürfnisse der gegebenen Aufgabenstellung hin kann die Existenz der benötigten hybriden Expertensystemschale auf Jahre gesichert werden. Diese Betrachtungen sollen exemplarisch anhand der KI-Werkbank Babylon, Version 2.3 [P15, S2, S3, X9] umgesetzt werden.

Babylon ist in der Programmiersprache CommonLISP [P12-P14] geschrieben. Daher wird weiterhin ein LISP-Interpreter benötigt. Neben käuflichen LISP-Interpretern wie Allegro CL [P8, P19] gibt es auch eine Anzahl auflagenfrei verfügbarer Interpreter wie CLISP [P20] und GCL [P21], die für unterschiedliche Betriebssysteme existieren.

Babylon kommuniziert mit dem Benutzer ausschließlich über Ein- und Ausgabedatenströme. Eine Volltextunterstützung und Mehrfenstertechnik sind nicht vorgesehen, da diese unter den verschiedenen Betriebssystemen unterschiedlich realisiert werden und auch die verschiedenen LISP-Interpreter nur zueinander inkompatible Möglichkeiten der Ausgestaltung von Benutzerschnittstellen bieten. Aus diesem Grund verzichtet Babylon auf eine Wissensingabeschnittstelle. Stattdessen ist es in der Lage, LISP-Dateien zu laden, die die Wissenskonstrukte einer Wissensbasis beinhalten. Damit erweist es sich insbesondere als leistungsfähige Inferenzmaschine, die um die grafischen Möglichkeiten einer durch die Wissensbasis flexibel ge-

staltbaren Benutzerschnittstelle zur Darstellung von Grafiken und Dialogführung durch Maskentechnik, sowie eine interaktive Wissensingenieurschnittstelle in Mehrfenstertechnik zu erweitern ist.

Derzeit existiert noch immer eine große Zahl etablierter Softwareanwendungen, die nur über eine Benutzerschnittstelle für Textbildschirme verfügt. Der im Kapitel 3 vorgeschlagene Weg läßt sich verallgemeinert nutzen, um mit nur geringem Programmieraufwand einen Großteil dieser betroffenen Anwendungen um die Möglichkeiten einer zeitgemäßen grafischen Mensch-Maschine-Kommunikation in Mehrfenstertechnik zu erweitern.

## **2.4 Ressourcenbedarf**

In Kapitel 1 sind einleitend Mindestanforderungen an Hard- und Software gestellt worden, die den Einsatz eines Expertensystems zur Transformatorauslegung in einer Schaltnetzteilentwicklungsabteilung erst ermöglichen.

- Es soll nur eine bereits vorhandene Rechnerhardware verwendet werden. Beispielsweise kann dies der vorhandene CAD-Arbeitsplatz für die Leiterplattenentwicklung sein, noch geeigneter wäre der eigene Arbeitsplatz-PC, also Systeme im unteren bis mittleren Leistungsbereich der Datentechnik.
- Damit ist auch eine Festlegung hinsichtlich des zugrundeliegenden Betriebssystems getroffen worden. Auf den benannten Rechnern werden bevorzugt Microsoft-Betriebssysteme oder UNIX-Derivate zum Einsatz kommen.

Um 1990, als mit einer Implementierung dieses Expertensystems begonnen wurde, war der typische Arbeitsplatz-PC mit einem Intel-80286-Prozessor, maximal 2 MB Hauptspeicher und monochromem Grafiksysteem ausgestattet. Als Betriebssystem kam MSDOS 3.1 zum Einsatz. Ein derartiger Rechner kam für den Betrieb eines derartig umfangreichen Expertensystems nicht in Frage. Als CAD-Arbeitsplatz dagegen wurde zumeist eine leistungsstarke UNIX-Graphic-Workstation mit hochauflösendem Vollfarbgrafiksysteem verwendet.

Heute sind die hardwareseitigen Grenzen zwischen Arbeitsplatz-PC und Graphic-Workstation verwischt. Das Microsoft-Betriebssystem Windows konkurriert mit UNIX und dessen netzwerkweit operierenden Fenstersystem X11 sowie der OSF/Motif-Fensterbibliothek, die dem X11-



Fenstersystem ein von Windows gewohntes Erscheinungsbild gibt [U2-U5, U13-U16]. Das Betriebssystem VMS [U7] kommt hier nicht zum Einsatz.

Daher wird der Lösungsvorschlag zur Ergänzung einer grafischen Benutzerschnittstelle zunächst nur im UNIX-Umfeld dargelegt. Ausblickend werden aber auch Hinweise für eine Umsetzbarkeit unter Windows-Betriebssystemen gegeben.

Das zu erstellende Expertensystem benötigt zum Ablauf einen LISP-Interpreter, erweitert um die Funktionalitäten Inferenzmaschine und grafische Benutzerschnittstelle, sowie die eigentliche Wissensbasis. Allein für den LISP-Interpreter CLISP samt Inferenzmaschine Babylon ohne geladene Wissensbasis ergab sich unter dem UNIX-Betriebssystem SunOS ein Verbrauch von virtuellem Arbeitsspeicher in Höhe von 2 MB. Dazu kommen diverse UNIX-Systemdienste, der X11-Server und diverse Grafikbibliotheken. Ein Speicherausbau von mindestens 16 MB RAM, besser 32 MB, ist für eine nicht mit weiteren Aufgaben belastete Graphic-Workstation angebracht. Bei Verwendung des Interpreters Allegro CL steigt der oben genannte Verbrauchswert von 2 auf 20 MB.

Mehrfenstertechnik stellt desweiteren hohe Anforderungen an das Grafiksubsystem. Verschiebeoperationen, sowie das Neuzeichnen einzelner Bildbereiche beispielsweise nach der Aufhebung von Verdeckungen sollen in einem für den Benutzer akzeptablen Zeitraum im Zehntelsekundenbereich abgeschlossen werden.

Schließlich stellt ein Expertensystem auch hohe Anforderungen an die zentrale Prozeßeinheit. Der hohe Abstraktionsgrad von Wissenskonstrukten benötigt einen höheren Interpretationsaufwand als bei Systemen, die beispielsweise von festgelegten Entscheidungsbäumen ausgehen dürfen. Die Bearbeitung einer Regelmenge mit einer großen Zahl Regeln führt dazu, daß ein Großteil der Rechenzeit zum Prüfen von Regeln verwandt wird, die nicht zur Ausführung gelangen. Dadurch können leicht störende Wartezeiten entstehen.

## **2.5 Terminalanwendungen und ereignisorientierte Anwendungen in Mehrfenstertechnik**

### **2.5.1 Interaktive Grafik**

Grafische Computerarbeitsplätze unterscheiden sich von Textterminals dadurch, daß mehr Möglichkeiten der Ein- und Ausgabe für die Mensch-Maschine-Kommunikation bereitstehen.

Textterminals bestehen aus einer Schreibmaschinentastatur, dem Keyboard, für die Eingaben und einem Bildschirm oder Display, auf dem im allgemeinen 25x80 oder 25x132 Zeichen dargestellt werden können. VT100 ist eine mögliche Festlegung des Leistungsumfangs für derartige visuelle Terminals. Visuell steht dabei für die Möglichkeit, nicht mehr nur einen reinen Ausgabedatenstrom - wie beispielsweise mit einem Drucker - darstellen zu können, sondern die Positionierung von Zeichen frei auf der Bildschirmmatrix wählen zu können.

Grafikfähige Terminals bieten neben der Tastatur zusätzlich ein Zeigergerät. Im allgemeinen ist dies eine mit 1 - 3 Knöpfen versehene Maus, auch Pointer genannt, oder ein Trackball. Für zweidimensionale CAD-Anwendungen stehen oft auch ein Graphic-Tablett, ein Joy-Stick oder ein Light-Pen zur Verfügung [U12]. Bei allen Geräten handelt es sich um grafische Eingabegeräte, die absolute oder relative Geometrieinformationen an den Rechner weitermelden. Diese Information wird - im Zusammenhang mit dem Status der Knöpfe - interpretiert als Lage im darstellbaren Bereich der Bildebene. Bei dreidimensionalen CAD-Anwendungen kann ein Werkstück beispielsweise auch als Drahtmodell dargestellt werden, das im Raum beliebig drehbar ist. Hierfür können spezielle Eingabegeräte auch räumliche Geometriedaten übermitteln. Dies kann einfach nur eine Lageposition im in der Bildprojektion dargestellten Raumstück sein. Es können darüberhinaus aber auch weitere Freiheitsgrade wie Zoom und Drehungen entlang der zu den drei Raumkoordinatenachsen parallelen Rotationsachsen sein.

Als Ausgabemedium dienen auch hier Bildschirme mit jeweils einer Matrix von Bildpunkten, Pixel genannt, wobei hier jedes Pixel einzeln ansteuerbar ist. Durch gezieltes Setzen bestimmter Pixel der Bildmatrix kann dem Betrachter der Eindruck vermittelt werden, es würde ein Buchstabe dargestellt. Diese Buchstaben sind dabei im Gegensatz zu den Textterminals in ihrer Größe veränderbar und innerhalb des darstellbaren Bereichs pixelgenau positionierbar. Zusätzlich können hiermit beispielsweise diagonale Linien und Ellipsen dargestellt werden. Bild 2.9 zeigt eine mögliche Ausstattung eines grafischen Rechnerarbeitsplatzes.

Die Rasterung der Bildfläche kommt von dem vom Fernsehen übernommenen Prinzip, daß der Bildaufbau zeilenweise geschieht. Es sind auch Displays denkbar, die wie bei einem Oszilloskop den Strahl der Elektronenkanone wahlfrei über die Mattscheibe lenken. So können Konturen frei von Rasterungsungenauigkeiten dargestellt werden. Buchstaben werden als Polygonzüge dargestellt.

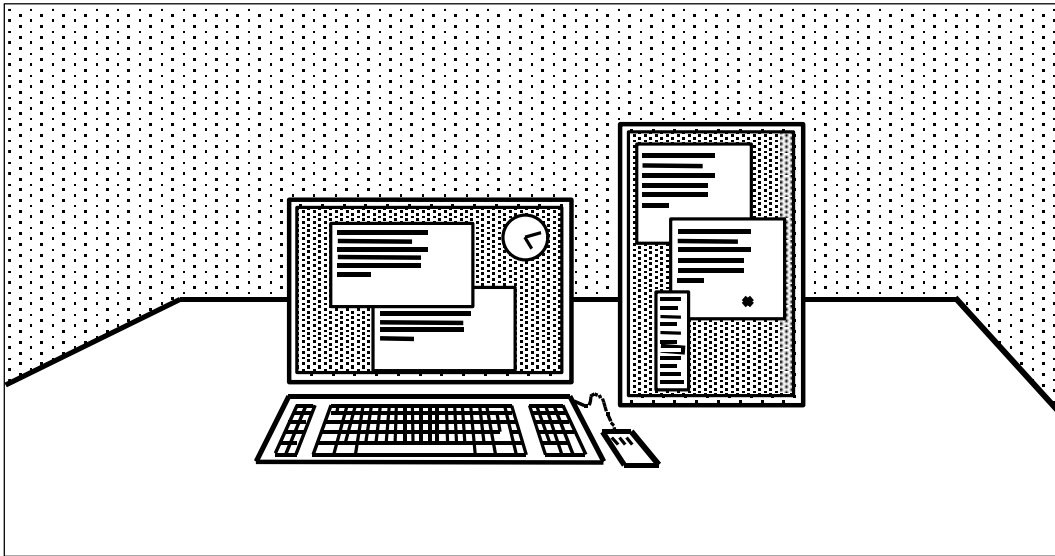


Bild 2.9: Ein grafischer Arbeitsplatz mit 2 Bildschirmen, Tastatur und Maus

Mit zunehmender Pixeldichte von Rasterdisplays relativiert sich dieser Vorteil. Die flächigen Darstellungen moderner grafischer Oberflächen führen dazu, daß auch wahlfrei positionierbare Displays diese Flächen streifenweise füllen müssen. Der höhere Aufwand zur Ansteuerung der Elektronenkanone rechnet sich gegenüber gängigen Rasterdisplays nicht mehr.

### 2.5.2 Programmfluß von Terminalanwendungen

Der typische Programmfluß von Anwendungen, die nur Ein- und Ausgaben auf Textterminals nutzen, ist in Bild 2.10 dargestellt. Der Programmablauf einer Anwendung verläuft ohne Beachtung des Zustands der Ein- und Ausgabegeräte, bis eine Benutzerreaktion erforderlich wird.

Die Anwendung springt dazu in die Routine „hole Zeichen“. Diese besteht aus einer Schleife, in der eine bestimmte Speicherstelle überwacht wird. Beispielsweise könnte dies ein Füllstandszeiger für den Tastaturpuffer sein. Der Druck des Benutzers auf eine Taste bewirkt einen Tastaturinterrupt. Eine Unterbrechungsbehandlungsroutine wird ausgelöst, die das dem Tastendruck entsprechende Zeichen in den Tastaturpuffer schreibt und die Füllstandsmarken anpaßt. Dieses von der Hardware ausgelöste Ereignis erfolgt asynchron zum Programmfluß [G4]. Die Anwendung kann nun das Zeichen aus dem Puffer auslesen, ihrerseits die Füllstandsmarken anpassen und die Eingaberoutine verlassen. Der Puffer kann auch in der Tastaturhardware realisiert sein. Dann ist keine spezielle Unterbrechungsverarbeitung notwendig. Das Auslesen des Hardwarepuffers nennt man Polling.

Auch bei der Routine „Zeichen schreiben“ kommuniziert die Anwendung mit der Hardware des Ausgabegerätes über einen Puffer. Dieser wird aber von der Anwendung befüllt und von der Hardware geleert. Hierzu wird im Pollingbetrieb gearbeitet. Die Anwendung wartet darauf, daß der Puffer soweit geleert worden ist, daß wieder Zeichen angenommen werden können.

### 2.5.3 Programmfluß ereignisorientierter Anwendungen

Grafikanwendungen besitzen mehrere Eingabegeräte, die sie gleichermaßen zu überwachen haben. Anstelle der Routine „Zeichen holen“ treten demgemäß die Routinen „Warte auf Ereignis“ und „Ereignis abarbeiten“. Zunächst werden die Ein- und Ausgabegeräte initialisiert und die gewünschten Verhaltensweisen festgelegt, die auf bestimmte Ereignistypen erfolgen sollen. Danach verzweigt der Hauptprogrammfluß in eine Endlosschleife, in der auf das Eintreten von Ereignissen gewartet wird. Die vorab zugeordneten Behandlungsroutinen werden nach Auswertung des Typs eines stattgefundenen Ereignisses abgearbeitet.

Entsprechend Bild 2.11 werden die Hardwarepuffer der Eingabegeräte selbst oder deren durch ihre Unterbrechungsbehandlungsroutinen befüllten Eingabepuffer der Reihe nach abgefragt. Bereits darin verbirgt sich eine Priorisierung von Eingabegeräten. Je nach Ereignistyp verzweigt die Routine „Ereignis abarbeiten“ in eine entsprechende in der Anwendung vorgesehene Behandlungsroutine. Dabei ist auch das Eingabegerät Wecker (Timer) zu berücksichtigen.

Der Hauptprogrammfluß von Grafikanwendungen muß **fragmentiert** in den unterschiedlichen Behandlungsroutinen untergebracht werden. Dort wird je nach Fortschritt der Anwendung in andere Fragmente verzweigt. Der Programmablauf ist **ereignisorientiert**. Terminalanwendungen kennen dagegen nur den Ereignistyp „alphanumerisches Zeichen vorhanden“. Eine Ereignisauswertungs- und Behandlungslogik ist nicht vorgesehen. Der Hauptprogrammfluß muß nicht fragmentiert werden.

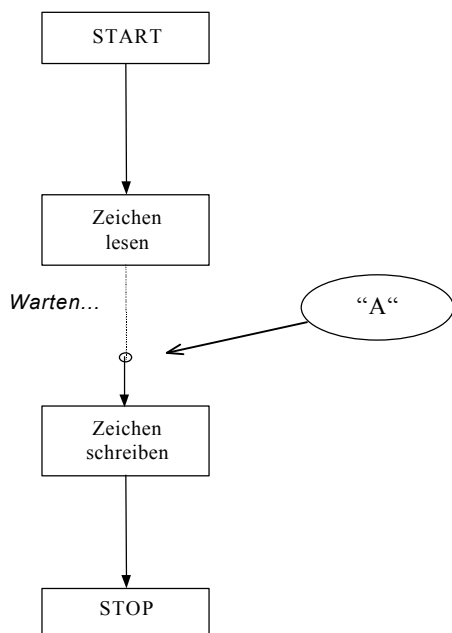


Bild 2.10: Stetiger Ablauf eines Programms für Textterminals

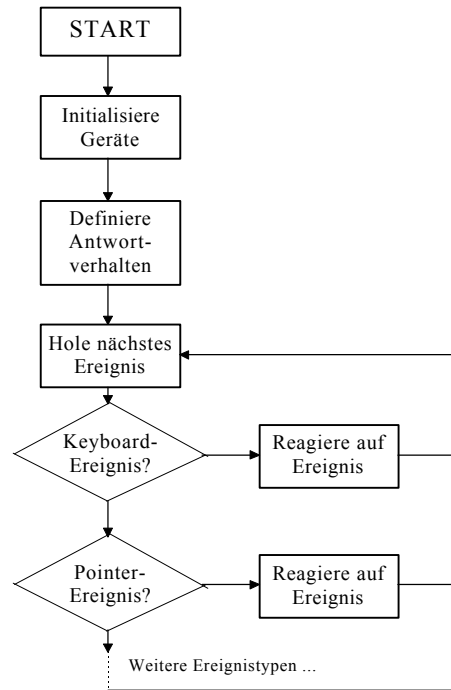


Bild 2.11: Fragmentierter Programmfluß einer Anwendung, die mehrere Eingabegeräte überwacht

## 2.5.4 Ereignisorientierte Programmierung

Ereignisorientierte Anwendungen in Mehrfenstertechnik müssen darüberhinaus mit der Aufhebung von Verdeckungen durch andere Fenster rechnen. Die Anwendung belegt nicht mehr den gesamten Bildschirm als einheitliche Zeichenfläche, vielmehr stellt jedes Fenster eine eigene Zeichenfläche dar. Die einzelnen Fenster können dabei wie Blätter auf einem Schreibtisch teilweise übereinanderliegen, sodaß es zu Verdeckungen von Fensterinhalten kommt. Dort darf die Anwendung auch dann nicht zeichnen, wenn die gesamte Fensterfläche innerhalb der Bildschirmfläche liegt. Das Verhindern von Zeichenoperationen außerhalb zulässiger Bereiche heißt Clipping.

Unter multitaskingfähigen Umgebungen können die Verdeckungsänderungen durch andere Fenster nicht mehr ausschließlich von der Anwendung selbst vorhergesehen werden, da diese nicht alle Fenster kontrolliert. Neben den geräteabhängigen Ereignistypen existieren weitere Ereignistypen, auf die die Anwendung zu reagieren hat. Weiterhin muß eine Zuordnung der Eingabegeräte zu einem Fenster erfolgen. Das Fenster, das gerade den Eingabefo-

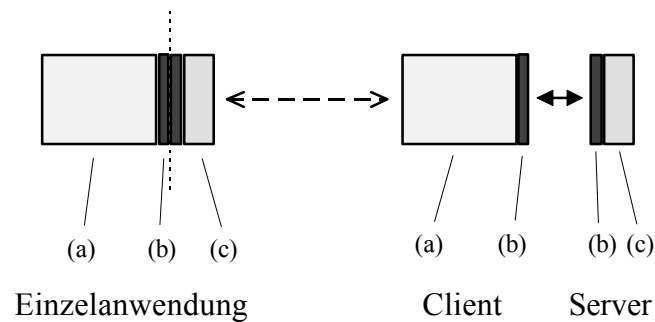
kus besitzt, muß nicht notwendigerweise auch das oberste Fenster im Stapel sein. Dies bedeutet weitere Ereignistypen, nämlich das Erhalten und das Verlieren des Fokus.

Da sich diese Fragmentierung für alle Arten von grafischen Anwendungen wiederholt, empfiehlt es sich, Funktionenbibliotheken bereitzustellen, die einen einheitlichen Weg der Ereignisbehandlung ermöglichen. Auf unterster Ebene empfiehlt sich die Einrichtung eines FIFO-Speichers, Ereigniswarteschlange oder Eventqueue genannt.

Die Ausgaben der unterschiedlichen Geräte werden dort in einem einheitlichen Datenformat zwischenspeichert. Jedesmal, wenn die Unterbrechungsbehandlungsroutine eines Geräts neue Daten sichert, werden die Art des Ereignisses, die Geometrieinformation, sowie ein Zeitstempel in einem Datensatz vermerkt. Bei dreidimensionalen Eingabegeräten beschreibt ein Ereignisdatsatz also einen fest definierten Punkt in einem virtuellen Raum-Zeit-Kontinuum der Anwendung [U12, U14]. Dies setzt voraus, daß bei keinem Gerät Pollingbetrieb vorgesehen ist.

Auch die Wechselwirkung mit anderen Anwendungen kann hierüber vereinheitlicht werden. Die Anwendungen kommunizieren miteinander nur indirekt über eine Mittlerebene. Soll ein Fenster komplett sichtbar werden, ist dies bei der Mittlerebene zu beantragen. Diese kennt die Positionen und Ausdehnungen aller Fenster, sowie deren Lage im Fensterstapel. Sie informiert die Gesamtheit der Anwendungen durch das Versenden von Ereignissen und genehmigt so letztendlich den Vorgang. So kann ein Fenster beispielsweise immer an oberster Stelle im Stapel plaziert werden. Nachdem ein derartiger Antrag der Anwendung bewilligt worden ist, wird die Mittler-ebene gleichartige Anfragen anderer Fenster nicht mehr bewilligen.

Neben der standardisierten Ereigniswarteschlange empfiehlt sich eine vereinheitlichte Reaktion auf bestimmte Ereignistypen. Ähnlich der hardwarenahen Zuordnung von Unterbrechungsbehandlungsroutinen kann eine Einsprungsadresse zur Ereignisbehandlungsroutine hinterlegt werden. Für derartige Routinen existiert dann eine einheitliche Argumentliste, die mindestens einen Verweis auf den Datensatz des zu verarbeitenden Ereignisses beinhaltet. So kann auch eine Trennung der hardwarenahen Treiberroutinen von den Programmierschnittstellen für Anwendungsentwickler vorgenommen werden. Die Verwendung von Geräten anderer Hersteller erfordert maximal Änderungen dieser systemnahen Routinen. Die Anwendung selbst ist davon nicht betroffen.



- (a) = Applikation
- (b) = Grafik-Bibliotheksfunktionen
- (c) = Routinen zur Ansteuerung der Display-Hardware

Bild 2.12: Entwicklung der Aufgabenverteilung in grafischen Client-Server-Anwendungen

Unter dem X11-Fenstersystem, das für die Derivate des UNIX-Betriebssystems entwickelt worden ist, ist darüberhinaus eine Aufgabenverteilung nach dem Auftragnehmer-Auftraggeber-Prinzip (Client-Server-Prinzip) umgesetzt worden. Die Hardware kann nur noch indirekt über einen Grafikservers, den X-Server, angesprochen werden. Die Anwendungen (Clients) beschreiben dem Server mittels eines Protokolls, das über Interprozesskommunikationskanäle gesandt wird, in welcher Weise er die Geräte zu beeinflussen hat. Andererseits ordnet der X-Server die Vielzahl der Ereignisse den unterschiedlichen Anwendungen zu. Die rechte Hälfte von Bild 2.12 verdeutlicht dieses Prinzip. Anwendungen, die unter den Fensterbetriebssystemen der Firmen Microsoft oder Apple ablaufen, führen eine derartige Trennung nicht durch (Bild 2.12 links).

Aufgrund der Abspaltung der hardwarenahen Routinen von denen der Anwendung sowie der bidirektionalen Einrichtung von Kommunikationskanälen können Server und Client auch auf verschiedenen Rechnern im Netzwerk ablaufen. Durch die Protokollfestlegung ist eine exakte Kenntnis der Displayhardware beim Client nicht mehr erforderlich.

Der X-Server übernimmt die Mittlerrolle für die Kommunikation zwischen den Anwendungen. Die Verwaltung des Fensterstapels sowie der Positionen der Fenster auf dem Bildschirm wird im allgemeinen einer speziellen Anwendung zugeordnet, dem Windowmanager. Pro Display ist maximal eine derartige Anwendung zulässig [U3, U4, U13, U14].

Unter den Fensterbetriebssystemen von Microsoft und Apple sind die Grafikbibliotheken fester Bestandteil des Betriebssystemkerns. Dieser übernimmt auch die Mittlerrolle [G5, G6].

### **2.5.5 Aufwandsabschätzung bei der Umstellung von Terminalanwendungen auf Mehrfenstertechnik**

Anwendungen für Fensterbetriebssysteme müssen auf externe Ereignisse reagieren, wie die Aufhebung von Verdeckungen. Wird dies unterlassen, kann die korrekte Darstellung der Anwendung nicht gewährleistet werden.

Modular programmierte Anwendungen besitzen eine geringe Zahl von Routinen, über die die Textein- und Ausgaben abgewickelt werden. Damit ist der Aufwand für die Umstellung dieser Funktionalitäten als gering zu bezeichnen.

Darüberhinaus könnten - wie beim Tastaturpuffer der Terminalanwendungen - im Rahmen dieser Ein- und Ausgaben die Ereigniswarteschlangen geleert werden. Dazu ist die Ereignisschleife gemäß Bild 2.11 so zu modifizieren, daß diese bei Nichtvorliegen von Ereignissen wieder verlassen wird. Eine derartige Routine „arbeite alle aufgelaufenen Ereignisse ab“ kann vor und nach dem Aufruf der eigentlichen Ein- und Ausgaberroutinen durchlaufen werden. Unter der Annahme, daß die Anwendung selbst nicht nebenläufig ist [G4], können zumeist die grafischen Ereignisse, die die Ein- und Ausgaben auslösen, noch abgearbeitet werden, bevor die Anwendung sich wieder auf ihre eigentlichen Aufgaben konzentriert.

Wenn dann aber über Minuten hinweg weder Ein- noch Ausgaben vorgesehen sind, kann eine konsistente Darstellung der Fensterinhalte nicht mehr gewährleistet werden. Je länger dieser Zeitraum ist und je mehr sonstige Aktivitäten auf dem System stattfinden, desto störender wird dies vom Benutzer empfunden. Dies kann bei Datenbankanwendungen der Fall sein, wenn eine aufwendige Suche über dem Datenbestand angestoßen worden ist. Dieser Vorgang ist für die Ereignisabarbeitung nur schwer unterbrechbar, da er innerhalb einer einzelnen SQL-Anweisung liegt. Diese beginnt mit dem Absetzen der Befehlszeichenkette an den Datenbankserver und endet mit dem Empfang der Ergebnisse von diesem [P7]. Aufgrund der Verarbeitungsgeschwindigkeit und der Größe des Datenbestands müßte vorab eine Abschätzung gemacht werden, in wieviele Teilsuchen der Vorgang unterteilt werden muß. Auch eine Unterbrechung pro untersuchtem Datensatz wäre denkbar, setzt aber einen Eingriff in die Bibliotheksfunktionen für eingebettetes SQL voraus. Dieser Weg ist daher wenig erfolgversprechend.



Die Inferenzmaschine der Expertensystemschaale Babylon ist ebenfalls sehr umfangreich. Sie ist modular programmiert. Es existieren daher nur wenige Routinen für Textausgaben, sowie zur Erfassung von Benutzerreaktionen. Andererseits kann die Bearbeitung eines Regelwerks ebenfalls mehrere Minuten dauern und damit über der Akzeptanzschwelle des Benutzers liegen. Eine Unterbrechung bei bestimmten Haltepunkten, beispielsweise vor jeder Regelprüfung, ist denkbar. Aber auch Regeln können in ihren Bedingungs- und Aktionsteilen Routinen haben, deren Abarbeitung lange dauert. Der vorgeschlagene Lösungsweg bleibt unsicher. Es existieren zwei sichere Alternativen:

1) Restrukturieren der gesamten Anwendung.

Dazu muß der bislang stetige Programmfluß gemäß Bild 2.10 fragmentiert werden und mit zusätzlichen Kontrollmechanismen zur Sicherung des ursprünglichen Flusses ausgestattet in den Ereignisabarbeitungsroutinen untergebracht werden. Eine Softwarepflege der eigentlichen Anwendung wird dadurch nicht notwendigerweise erleichtert. Für umfangreiche Anwendungen stellt sich dieser Weg als sehr arbeitsintensiv dar.

2) Nebenläufige Aufgabenverteilung.

Wenn das zugrundeliegende Betriebssystem Nebenläufigkeiten geeignet unterstützt, kann die gesamte Grafikaktivität der Anwendung in einen nebenläufigen Zweig ausgelagert werden [G4]. Ein Ansatz für preemptive Multitaskingsysteme wie UNIX wird im Kapitel 3 am Beispiel der Expertensystemschaale Babylon vorgestellt.

## 2.6 Zusammenfassung

In diesem Kapitel wurde zunächst belegt, daß es eine umfangreiche Aufgabe darstellt, eine Assistenzsoftware für die Unterstützung eines Entwicklungsingenieurs bei der Auslegung von Mittelfrequenztransformatoren zu erstellen. Eine komplette Neuentwicklung ist nicht sinnvoll, weil es eine Vielzahl von Softwarewerkzeugen gibt, bei deren Inanspruchnahme die Gesamtentwicklungszeit deutlich reduziert werden kann. Hybride Expertensystemschaalen haben sich dabei als besonders geeignet herausgestellt.

Eine weitere Forderung an die Assistenzsoftware ist Grafikfähigkeit. Die Mensch-Maschine-Kommunikation soll sich nicht nur auf Maskentechnik beschränken, Grafiken und Diagramme sollen den Dialog bereichern können.

Es existiert eine große Zahl von Softwareanwendungen, die sich über Jahre hinaus bewährt haben. Sie sind für den Ablauf in Textterminalumgebungen entwickelt worden. Heute sinkt die Akzeptanzschwelle für derartige Programme, da die Präsentation als veraltet empfunden wird. Um diese Anwendungen attraktiv zu erhalten, ist eine grafische Benutzeroberfläche in Mehrfenstertechnik geeignet zu ergänzen. Es konnte aufgezeigt werden, daß es dazu nicht reicht, die vorhandenen Quelltexte dieser Anwendungen um einige Grafikanweisungen zu ergänzen. Vielmehr ist der gesamte Programmfluß zu fragmentieren und reorganisieren.

Dies gilt auch für die Expertensystemschale Babylon, für die im folgenden Kapitel 3 exemplarisch ein Lösungsvorschlag erarbeitet wird, der eine derart umfangreiche Reorganisation der umfangreichen Quelltexte überflüssig macht. Dieses Konzept läßt sich auf eine Vielzahl anderer Terminalprogramme anwenden, für die eine Neuimplementierung aufgrund der Größe nicht in Frage kommt.

### **3. Nebenläufige Erweiterung von Terminalanwendungen um eine grafische Oberfläche am Beispiel von Babylon**

Soll eine bestehende Software um ein komplexes Modul erweitert werden, sind Eingriffe auf allen Programmiererebenen denkbar. Diese reichen von betriebssystemnahen Routinen bis hin zur Ebene der Applikationsprogrammierung. Daher ist eine Analyse dieser Software in allen Bereichen unumgänglich.

Um bei der Erweiterung einer Terminalsoftware um eine grafische Bedienoberfläche die vorhandenen Quelltexte nicht fragmentieren und restrukturieren zu müssen, ist in Kapitel 2.5 ein Lösungsweg vorgeschlagen worden, der die Fähigkeit des Betriebssystems voraussetzt, Rechenprozesse nebenläufig abarbeiten zu können. Dazu muß zuerst festgestellt werden, ob und in welchem Umfang das zugrundeliegende Betriebssystem Nebenläufigkeit unterstützt. Daraufhin muß eine Aufgabenverteilung unter den Rechenprozessen festgelegt werden, die wiederum den Bedarf an Prozeßsynchronisation und Interprozeßkommunikation bedingt. Gegebenenfalls ist die Aufgabenverteilung zu überdenken, um nicht mit den Möglichkeiten des Betriebssystems in Konflikt zu geraten.

Daher werden in diesem Kapitel zunächst gängige Betriebssysteme betrachtet. Daraufhin wird ein konzeptioneller Grundriß für eine Problemlösung erarbeitet. Dieser wird schließlich am Beispiel der Schale Babylon individuell verfeinert.

#### **3.1 Nebenläufigkeit in gängigen Betriebssystemumgebungen**

Bedingt durch ihre Konzeption laufen ereignisorientierte Anwendungen sequentiell ab. Das heißt, sie sind auch auf einem Singletasking-Betriebssystem wie MSDOS (Microsoft Disk Operating System) funktionsfähig. Unter MSDOS, das um 1990 noch das meistverwendete Betriebssystem für PCs war, ist der Prozessor exklusiv einer einzigen Anwendung zugeordnet. Diese Anwendung kann eine weitere Anwendung starten, ist dann selbst aber inaktiv, bis letztere terminiert. Nebenläufigkeit ist nicht vorgesehen.

Als eine Weiterentwicklung aus dem gleichen Hause versteht sich das Fensterbetriebssystem Windows 3.1. Es wird als eine MSDOS-Anwendung gestartet und kann auch so beendet werden, daß sich der Benutzer an der Eingabeaufforderung von MSDOS wiederfindet. Unter Windows können mehrere Grafikanwendungen in Mehrfenstertechnik nebenläufig abgearbeitet werden. Allerdings entscheidet jede Anwendung selbst, wann sie den Pro-

zessor wieder freigibt. Man spricht von kooperativem Multitasking [G5, G6]. Die Entscheidung über die Neuvergabe des Betriebsmittels Prozessor erfolgt nur bei Aufruf der Betriebssystemroutine „Bearbeite ein Ereignis“. Verzweigt eine Anwendung für eine gewisse Zeit nicht in diese Routine, beispielsweise, weil eine umfangreiche Berechnung durchgeführt wird, findet systemweit keine Ereignisbearbeitung statt. Nebenläufigkeit ist unter den gegebenen Umständen nur bei Einhaltung einer strengen Programmierdisziplin für alle unter diesem Fensterbetriebssystem laufenden Anwendungen möglich. Auch das Betriebssystem MacOS der Firma Apple verwendet kooperatives Multitasking.

Windows 95 ist die Nachfolgeversion von Windows 3.1. Neue Betriebssystemfunktionalität läßt zu, daß eine Anwendung in mehrere Threads unterteilt wird, die wiederum nebenläufig abgearbeitet werden [G5, G6]. Threads sind Rechenprozesse, die den gesamten Programm- und Datenbereich der Anwendung gemeinsam nutzen. Synchronisation über Variablen ist somit leicht möglich [G4]. Die Prozessorzuteilung innerhalb dieser Threads erfolgt präemptiv. Das heißt, den einzelnen Threads einer Anwendung wird nach einer bestimmten Zeit der Prozessor von einer Verwaltungseinheit (Scheduler) wieder entzogen. Andererseits funktioniert dieses Verfahren nur solange, wie keine für Windows 3.1 entwickelte Anwendung den Prozessor zugeweiht bekommt. Denn diese verfügt noch nicht über eine entsprechende Verwaltungseinheit, beziehungsweise besteht nur aus einem einzigen Thread. Auch im Umfeld dieses Betriebssystems kann daher Nebenläufigkeit nicht sichergestellt werden.

Das Betriebssystem UNIX ist in den siebziger Jahren in Zusammenarbeit zwischen dem Massachusetts Institute of Technology (MIT), den Bell-Laboratories und General-Electric entwickelt worden. Ab 1976 wurde an der University of California at Berkeley (UCB) eine eigene erweiterte Linie von UNIX Systemen entwickelt [U9, U10, U17]. Auch andere kommerzielle Anbieter brachten UNIX-ähnliche Betriebssysteme heraus, erweitert um hauseigene Besonderheiten [U1, U8, U10, U19]. UNIX war von Anfang an als ein Mehrbenutzerbetriebssystem konzipiert worden, das präemptives Multitasking verwendet, um unterschiedlichen Prozessen den Prozessor zuzuteilen. Die Prozeßzuteilung regelt der Scheduler nach einer prioritätenorientierten Vergabestrategie. Höherpriorisierte Prozesse werden bevorzugt, gleichwertige Prozesse der Reihe nach bedient. Jeder Prozeß wird für maximal die Dauer einer Zeitscheibe aktiviert und danach unterbrochen. Die Länge einer Zeitscheibe liegt im Millisekundenbereich. Dies bewirkt eine

pseudoparallele Abarbeitung unterschiedlicher Aufgaben. Der Begriff Prozeß unterscheidet sich von dem in [G4] definierten Rechenprozeß schon darin, daß jeder Prozeß einen eigenen Programm- und Datenbereich besitzt. Eine virtuelle Speicherverwaltung verhindert, daß sich die Programm- und Datenbereiche unterschiedlicher Prozesse überlappen können. Auch unter UNIX gibt es die Möglichkeit, einen Prozeß in mehrere Threads zu unterteilen.

Die UNIX-Derivate erweisen sich als stabiler als die Windows-Varianten, da die Zuteilung der Betriebsmittel im Betriebssystemkern erfolgt und nicht in der Anwendung. Nebenläufige Programme lassen sich unter UNIX sehr gut realisieren.

### **3.1.1 Prozessorzuteilung**

Bezogen auf die gestellte Aufgabe bedeutet der Einsatz eines Betriebssystems mit einer kooperativen Prozessorvergabestrategie, daß eine Vielzahl von Eingriffen in die Quelltexte der Software nötig ist, um Nebenläufigkeit zu gewährleisten. Die Eingriffe wären sehr gezielt mit dem Laufzeitverhalten der Anwendung zu koordinieren. Bei einem derart hohen Aufwand zur Sicherung der Nebenläufigkeit empfiehlt sich eine Restrukturierung der Software eher, da andere Anwendungen nicht notwendigerweise so kooperativ programmiert sind, wie es erforderlich wäre. Außerdem ist nach der Restrukturierung die Software viel leichter portierbar geworden. Nebenläufigkeit ist dann nicht mehr erforderlich.

Bei präemptiven Multitaskingbetriebssystemen wird der Prozessor in festen Zeitabständen neu zugeteilt. Für die Anwendungsentwicklung ist dies bezüglich geforderte verschärfte Programmierdisziplin nicht notwendig.

### **3.1.2 Speicherzuteilung**

Das Unterteilen einer Anwendung in mehrere Threads bedeutet auch, daß alle Threads Zugang zu allen Daten der Anwendung haben. In Hinblick auf die Prozeßsynchronisation erweist sich dies als ein Vorteil.

Am Beispiel der Schale Babylon zeigt sich, daß einerseits ein LISP-Interpreter zum Einsatz kommt. Der LISP-Interpreter soll dabei als ein geschlossenes System angesehen werden, um sich nicht von einem bestimmten Produkt abhängig machen zu müssen. Da die Threads auch den Programm-bereich gemeinsam nutzen, müßte die Nebenläufigkeit in den LISP-Interpreter integriert sein, was einer Spracherweiterung gegenüber CommonLISP

entspreche [P12-P14]. Damit wäre man mit der Ausgestaltung der grafischen Oberfläche auch an die Möglichkeiten des Interpreters gebunden.

Die Entkopplung der Adressräume der einzelnen Prozesse unter UNIX bedeutet einerseits einen Mehraufwand bei der Synchronisation und der Kommunikation, andererseits können so sehr unterschiedliche Programme betriebsystemunterstützt miteinander gekoppelt werden. Für Babylon heißt dies, daß der LISP-Interpreter nur sprachkonform verwendet werden muß. Ein weiteres Programm, das in einer anderen Programmiersprache entwickelt sein kann, erledigt die verbliebenden Aufgaben.

Unter Berücksichtigung der Leistungsfähigkeit der betrachteten Betriebssysteme empfiehlt sich der Einsatz eines UNIX-Derivats für die gestellte Aufgabe. Nicht zuletzt, wenn man bedenkt, daß mittlerweile auch eine Reihe von kommerziellen wie freien Implementierungen für PC-Hardware verfügbar sind.

### 3.2 Möglichkeiten der Synchronisation und Kommunikation

Unter UNIX wurden bereits sehr früh Möglichkeiten der Prozeßsynchronisation sowie der Interprozeßkommunikation, auch über ein Rechnernetzwerk hinweg mit Prozessen auf anderen Rechnern, bereitgestellt.

Der Scheduler ist als übergeordnete Instanz in der Lage, **Signale** an die einzelnen Prozesse zu versenden. Diese lassen sich am ehesten mit einem Hardware-Interrupt vergleichen. Der Prozeß sieht für jedes mögliche Signal ein Antwortverhalten in Form einer Signalbehandlungsroutine vor. Signale sind ein abstraktes Mittel, dem Prozessor besondere Systemzustände mitzuteilen, wie der Versuch eines Prozesses, einen Speicherbereich zu adressieren, der außerhalb des ihm zugeordneten virtuellen Speicherraums liegt. Ein Prozeß A kann aber auch einem Prozeß B ein Signal zukommen lassen. Da die Anzahl der unterschiedlichen Signalarten sehr begrenzt ist, ist damit zwar eine schnelle Synchronisierung möglich, der Datenaustausch muß aber mit anderen Mitteln bewerkstelligt werden.

Das UNIX-Dateisystem ist so organisiert, daß eine Datei, die ein Prozeß zum Schreiben geöffnet hat, nicht auch von einem anderen Prozeß zum Schreiben geöffnet werden kann. Dieser Mechanismus kann zur einseitigen Synchronisation genutzt werden. Da hierfür das langsame Dateisystem benötigt wird, sollte es nicht für häufige und zeitkritische Synchronisationsaufgaben verwendet werden. Der Prozeß wird beim Versuch, auf eine **verriegelte Datei** zuzugreifen, blockiert. Erst nach Aufhebung der Verriegelung wird der Prozeß fortgesetzt.

Der schnellste Datenaustausch wird dadurch ermöglicht, daß ein Bereich des Arbeitsspeichers (RAM) der virtuellen Speicherverwaltung entzogen und mehreren Prozessen als gemeinsamer Datenbereich (**sharable memory**) zugeordnet wird. Hierin lassen sich beispielsweise FIFO-Puffer (First In First Out) in Form von Ringpuffern aufbauen. Zur Synchronisierung muß der Prozeß ständig relevante Änderungen des Speicherinhalts überwachen (polling). Dies erfordert Rechenleistung. Gemeinsam genutzte Datenbereiche sollten nur in kleinen Mengen angefordert werden, weil es den Wirkungsgrad der Speicherverwaltung mindert. Hierbei handelt es sich um Speicher, der explizit an das Betriebssystem zurückgegeben werden muß im Gegensatz zum virtuellen Adressraum jedes Prozesses [U11]. Gemeinsam genutzte Datenbereiche entsprechen am ehesten einem Dual-Port-RAM in reinen Hardwarelösungen.

Eine klassische Form der Prozeßsynchronisierung sind **Semaphorvariablen**. Sie wurde Mitte der sechziger Jahre von E. W. Dijkstra entwickelt [G4, P11]. Mit ihnen läßt sich die korrekte Abarbeitung kritischer Abschnitte einer komplexen Aufgabe sicherstellen. Andererseits kann eine ungeschickte Verwendung dieses Mechanismus leicht eine Verklemmung hervorrufen. Dann warten die betroffenen Prozesse beliebig lange darauf, daß der andere die Freischaltung bewirkt.

**Pipelines** präsentieren sich anwendungsseitig als spezielle Dateien. Sie werden zweimal geöffnet, einmal nur zum Lesen und einmal nur zum Schreiben. Dahinter verbirgt sich ein im Speicherbereich des Systemkerns vorgehaltener Ringpuffer. Der Zugriff auf Pipelines erfolgt mit den Systemfunktionen für Dateien. Es gibt benannte und namenlose Pipelines. Benannte Pipelines erhalten zusätzlich einen Eintrag in den Verzeichnisbaum. Dies ist vorteilhaft, wenn ein anderer Prozeß das zweite Ende dieser FIFO öffnen will, der nicht durch Gabelung aus dem ersten hervorgegangen ist. Um den Systemspeicherbedarf in Grenzen zu halten, ist die Anzahl der Ringpuffer und damit die der Pipelines begrenzt.

**Sockets** bieten auf der Basis des Internet-Protokolls (IP) Zugang zum Rechnernetzwerk. Beispielsweise wird hierüber die Kommunikation zwischen dem X-Server und seinen Client-Anwendungen abgewickelt [U9, U10, U17].

### 3.3 Die Pseudoteletype-Gerätetreiber und die X11-Anwendung xterm

Im Unterschied zu den bislang aufgezählten Mechanismen existieren **Pseudoteletypes** nur unter dem Betriebssystem UNIX. Dies erklärt sich aus

der Entwicklungsgeschichte von UNIX. Pseudoteletypes stellen eine leistungsfähigere Variante von Pipelines dar. Pseudoteletypes (PTY) bilden das Verhalten von Teletypes (TTY) nach. Eine Teletype ist ein Gerät, hinter dem ein Textterminal vermutet wird, wie in Bild 3.1 dargestellt.

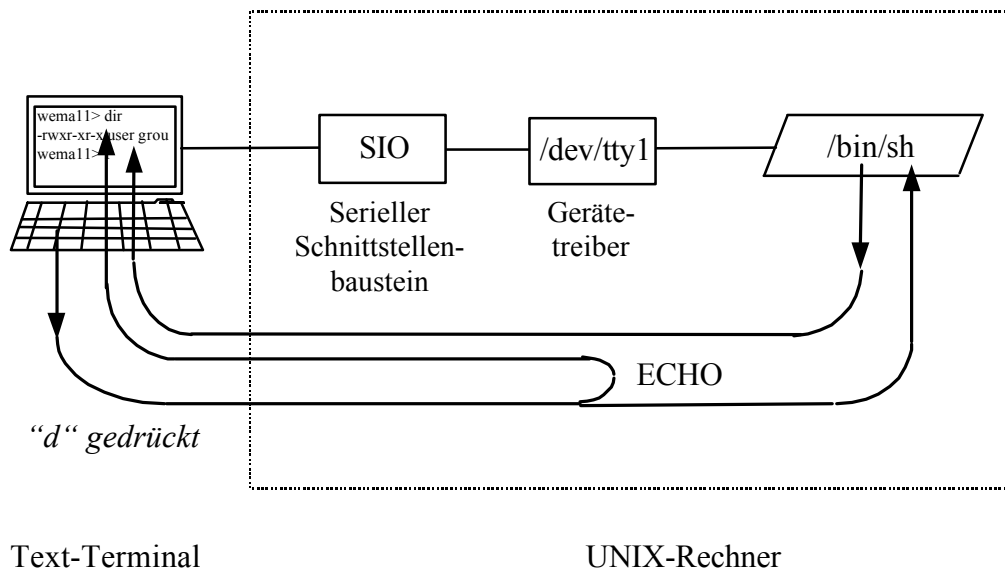


Bild 3.1: Mensch-Maschine-Kommunikation über eine Teletype

UNIX ist als Mehrbenutzersystem konzipiert worden. Demgemäß sollen mehrere Menschen gleichzeitig mit dem System kommunizieren können. Jeder Benutzer benötigt dazu einen eigenen Arbeitsplatz. Dies kann ein Textterminal sein, das über eine serielle Schnittstelle mit dem UNIX-Rechner verbunden ist. Die SIO-Hardware (Serial Input and Output Device) wird von einem Treiber bedient, der Bestandteil des Systemkerns ist. Standardeingabe und Standardausgabe des ablaufenden Programms, im Bild der Kommandozeileninterpreter /bin/sh, werden durch Öffnen einer Gerätedatei zum Lesen und Schreiben (im Bild /dev/tty1) mit dem im Kernel vorgehaltenen Gerätetreiber für das gewünschte Gerät (Device) verbunden.

Drückt der Benutzer die Taste „d“, wird der Code des Zeichens vom Terminal über die Schnittstelle an den Treiber übermittelt. Dieser sendet bei Betriebsart ECHO dieses Zeichen an das Terminal zurück, das daraufhin ein „d“ auf dem Display darstellt. In der RAW-Betriebsart wird dieses Zeichen dann sofort vom Treiber an den Prozeß weitergeleitet. In der COOKED-Einstellung dagegen werden solange Zeichen gepuffert, bis eine Eingabezeile komplettiert und die Wagenrücklaufaste (CR) gedrückt ist. Dann erst wird die gesamte Zeile weitergereicht. Wahlweise kann CR als CR, Neue-



Zeile (NL) oder CR und NL an den Prozeß weitergeleitet werden. Voreingestellt sind ECHO- und COOKED-Modus, sowie die Umwandlung von CR in CR-NL.

Ein Pseudoteletype besteht aus zwei Gerätetreibern (beispielsweise /dev/ttyp0 und /dev/ptyp0). Der ttyXY-Gerätetreiber ist konfigurierbar wie ein Teletypetreiber (RAW/COOKED und dergleichen) und bildet beispielsweise auch das ECHO-Verhalten nach. Anstelle des Zugangs zur Hardware stehen dagegen zwei Ringpuffer für den bidirektionalen Datenaustausch zum ptyXY-Gerät bereit. Ein anderer Prozeß öffnet das Gerät zum Lesen und Schreiben und kann nun beispielsweise das Verhalten des Textterminals emulieren.

Dies ist die Arbeitsweise der **X11-Anwendung xterm**. Mit ihr können für Textterminals geschriebene Programme auch unter dem X11-Fenster-system weiterbetrieben werden [U3, U11, U18]. Diese Anwendung initialisiert zunächst ihre Grafik, öffnet danach ein pty/tty-Pseudogerätepaar und zweigt danach den Nutzprozeß ab. Dessen Standardeingabe und Standardausgabe werden auf das tty-Pseudogerät umgelenkt. Bild 3.2 verdeutlicht die gegenüber Bild 3.1 abgewandelte Textein- und Ausgabe beim xterm.

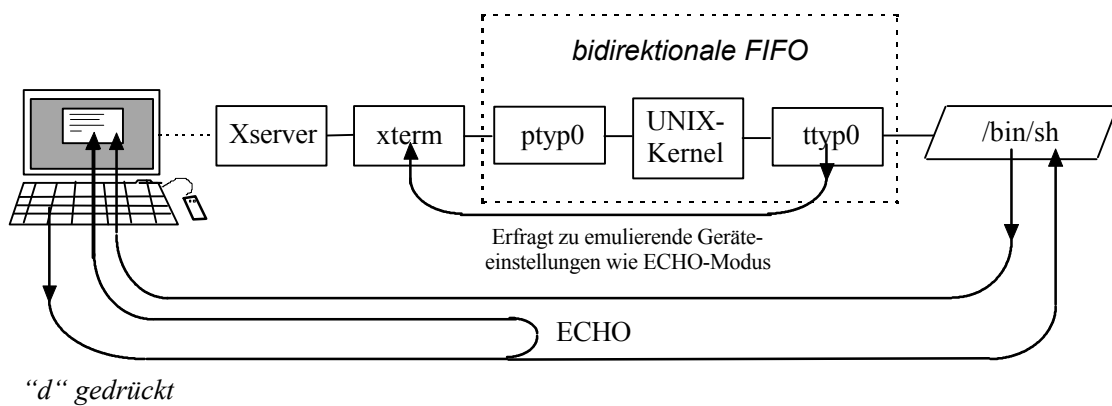


Bild 3.2: Mensch-Maschine-Kommunikation beim Terminalemulator xterm

Ebenfalls in der Vergangenheit begründet ist die Existenz der Betriebssystemfunktion **isatty**. Durch sie kann beispielsweise festgestellt werden, ob die Standardeingabe auf eine Teletype geleitet worden ist oder aber auf eine Datei oder Pipeline. Bei der Teletype kann davon ausgegangen werden, daß eine interaktive Sitzung mit einem Benutzer stattfindet, in den anderen Fällen findet ein Stapelbetrieb statt [G4]. Kommandozeileninterpreter, aber auch viele LISP-Interpreter machen davon Gebrauch. Beispielsweise schreiben die Interpreter im interaktiven Betrieb eine Eingabeaufforderung in die

Standardausgabe, bevor sie auf Eingaben warten. Im Stapelbetrieb terminiert der Interpreter, sobald keine Zeichen mehr von der Standardeingabe gelesen werden können.

### **3.4 Bestandsaufnahme für die Schale Babylon**

#### **3.4.1 Modularer Aufbau der Expertensystemschaale Babylon**

Die hybride Expertensystemschaale Babylon zeichnet sich - bedingt durch eine durchgehend objektorientierte Programmierung - darin aus, daß Interpretermodule unterschiedlicher Leistungsfähigkeit miteinander verschaltet werden können. Die Leistungsfähigkeit jedes einzelnen Expertensystems kann an den Leistungsbedarf der zugehörigen Wissensbasis angepaßt werden. Für jeden Wissensrepräsentationsformalismus steht ein eigener spezialisierter Wissensinterpreter zur Verfügung. Dabei handelt es sich um Frame-, Regel-, Constraint- und Prologinterpreter. Jeder dieser Interpreter steht in drei Ausbaustufen zur Verfügung. Die „Basic“-Variante stellt alle Grundfunktionalität zur Verfügung, die zum uneingeschränkten Betrieb notwendig sind. In den darauf aufbauenden Varianten „Mini“ und schließlich „Normal“ kommen Tracer- und Erklärungsmöglichkeiten hinzu. Darüberhinaus steht ein Freitextinterpreter zur Verfügung. Ein definierter Zugang zum sequentiell arbeitenden LISP-Bereich, der den Aufruf von LISP-Funktionen beispielsweise aus Regeln heraus ermöglicht, kann zugeschaltet werden. Schließlich existieren Module für die Mensch-Maschine-Kommunikation.

Diesen spezialisierten Einzelinterpretern ist als überlagerte Kontroll- und Koordinationsinstanz der sogenannte Metainterpreter vorgesetzt. Er kontrolliert den globalen Sitzungsablauf und verweist die einzelnen Anweisungen zur Behandlung an die jeweiligen Wissensinterpreter. Diese können - aufgrund des hybriden Ansatzes - bei der Bearbeitung wiederum auf Anweisungen stoßen, die ihnen fremd sind. Sie geben die betreffenden Teilanweisungen an den Metainterpreter zur geeigneten anderweitigen Zuweisung zurück und erhalten das benötigte Ergebnis der Interpretation später vom Metainterpreter zurück, um die begonnene Teilaufgabe fortführen zu können. Daher genügt es, wenn jeder spezialisierte Interpreter mit dem Metainterpreter kommunizieren kann. Von der Existenz der anderen Spezialisten erfährt er nichts. Bild 3.3 verdeutlicht die Architektur der KI-Werkbank Babylon [S2, S3].

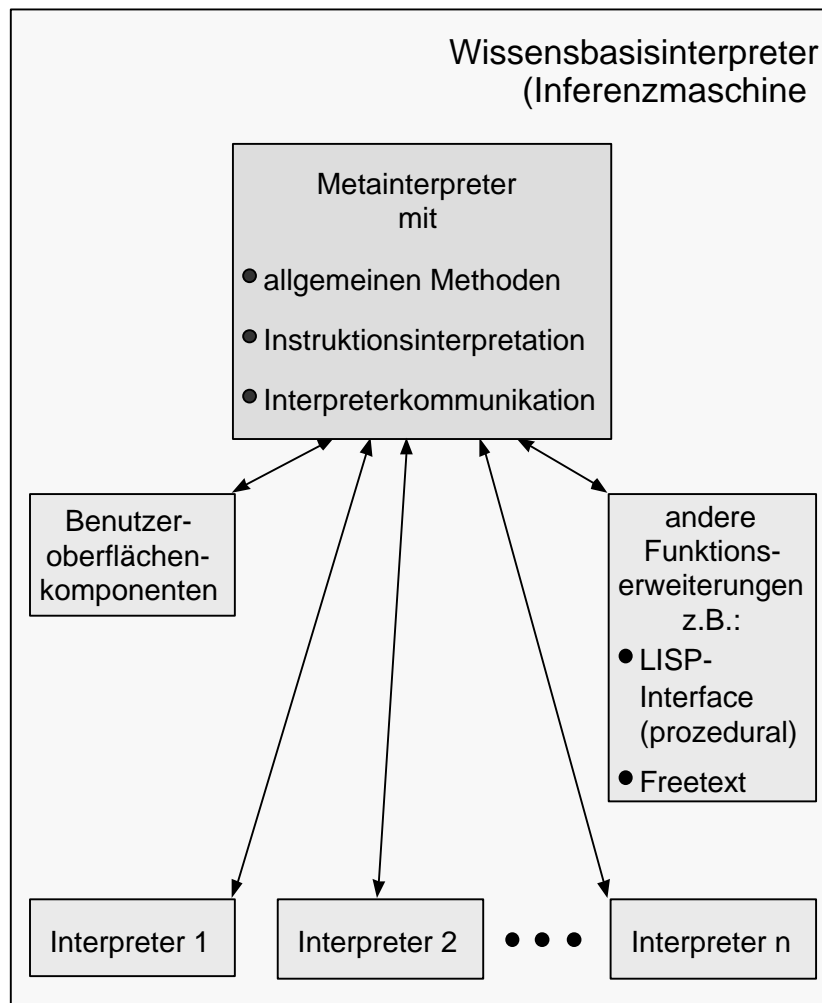


Bild 3.3: Die hybride Architektur der KI-Werkbank Babylon

### 3.4.2 Die Mensch-Maschine-Schnittstelle von Babylon

Babylon bietet neben den Wissensinterpretern auch ein Modul für die Mensch-Maschine-Kommunikation an. Dieses existiert ebenfalls in drei Ausprägungen. Das Basic-Interface-Mixin stellt die elementaren Ein- und Ausgabeoperationen zur Verfügung. Sämtliche Kommunikation wird hierüber abgewickelt. Es wird zwischen unterschiedlichen logischen Ausgabeströmen unterschieden, die aber voreingestellt alle auf den Standardausgabekanal umgelenkt und somit zusammengefaßt sind.

Neben dem bidirektionalen LISP Standardein- und Standardausgabekanal, darüber wird im Fehlerfall mit dem interaktiven Debugger des LISP-Interpreters kommuniziert, existieren ein bidirektionaler Sitzungsdialogstrom sowie Ausgabekanäle für Erklärungen und die Trace-Informationen des Regel-, Prolog-, Constraint- und Metainterpreters. So können beispielsweise

die jeweiligen Protokollausgaben in unterschiedliche Ausgabedateien umgelenkt werden. Dies ist bei der Erstellung eines Expertensystems vorteilhaft.

Das Mini-Interface-Mixin stellt - immer noch basierend auf dem Modell der Ein- und Ausgabeströme - die Möglichkeit zur Verfügung, menügeführt auszuwählen.

Das Normal-Interface-Mixin ist derzeit nur in der Version für den Apple Macintosh unter dem LISP-Interpreter Allegro CL [P8, P19] verfügbar. Es verwendet mehrere Textfenster, um die Informationen der Ausgabeströme geeignet darstellen zu können, und Auswahldialoge für die Systemeinstellungen und Benutzereingaben. Diese Lösung ist nicht portabel. Weiterhin entsprechen die grafischen Möglichkeiten der Macintosh Version nicht den Forderungen, aus der Wissensbasis heraus individuelle grafische Benutzerschnittstellen generieren zu können [S2, S3]. Ebenfalls fehlt eine Wissenseingabeschnittstelle.

### **3.5 Anforderungsprofil für eine Erweiterung der Schale Babylon**

Aufgrund der bisherigen Betrachtungen läßt sich zusammenfassend folgendes feststellen:

- Zur Laufzeit der Gesamtanwendung operieren 2 nebenläufige Programme, zwischen denen Kommunikationskanäle bidirektional eingerichtet worden sind.
- Eines der Programme ist die geringfügig modifizierte Originalanwendung.
- Bei dem anderen Programm handelt es sich um eine Grafikanwendung.
- Wenn die Grafikanwendung startet, legt sie zunächst ihr ereignisbedingtes Verhalten fest. Eine Klasse von Ereignissen ist dabei die Reaktion auf eingehende Nachrichten aus den Kommunikationskanälen.
- Da die Originalanwendung möglichst geringfügig modifiziert werden soll, kann die Grafikanwendung als erste gestartet werden, die Kommunikationskanäle einrichten, die Originalanwendung abspalten und in diesem Zusammenhang mit den jeweiligen anderen Enden der Kommunikationskanäle verbinden.

Über das bisher zusammengefaßte hinaus muß eine wechselseitige Benachrichtigung vom unerwarteten Terminieren des Partnerprogramms eingerichtet werden, um die Kommunikationskanäle abzubauen und die Gesamtanwendung stabil beenden zu können.

Zeitgemäße grafische Anwendungen besitzen ein Hauptfenster, von dem aus menügesteuert die unterschiedlichen Operationen, die die Anwendung ausmachen, initiiert werden können.

Insbesondere für eine aufgabengerechte Erweiterung der Schale Babylon bedeutet dies:

- Der Originalprozeß ist ein LISP-Interpreter, der als geschlossenes System angesehen werden soll, um sich nicht von einem bestimmten Produkt abhängig zu machen.
- Es werden - wie auch schon in den Quelltexten von Babylon - nur LISP-Kommandos verwendet, die auf der Mehrzahl der LISP-Interpreter verfügbar sind.
- Daher dürfen nur Anpassungen an den Quelltexten von Babylon vorgenommen werden.
- Idealerweise wären dazu nur geringfügige Eingriffe in das Basic-Interface-Mixin notwendig, also in die elementaren Ein- und Ausgaberroutinen.
- Für den LISP-Interpreter bedeutet dies insbesondere, daß zur Anbindung der Kommunikationskanäle nur die Dateioperationen Öffnen, Lesen, Schreiben, Schließen verwendet werden dürfen.
- Daraus ergibt sich wiederum, daß nur die bereits geöffnete Standardein- und Ausgabe, sowie benannte Pipelines in Frage kommen.
- Unter UNIX prüfen Interpreter, ob sie in der Stapelverarbeitung eingesetzt werden oder ob eine interaktive Sitzung stattfindet. Daher muß die Standardeingabe auf eine Pseudoteletype gelenkt sein.

Die Gesamtanwendung soll vom Hauptfenster aus kontrolliert werden. Dies beinhaltet das Laden von Wissensbasisdateien in die Expertensystemschale, das Speichern von Änderungen der Wissensbasis, sowie das Entfernen einer Wissensbasis aus der Expertensystemschale.

Da Babylon mehrere Wissensbasen gleichzeitig im Speicher halten kann und eine Wissensbasis zur Sitzungszeit eine andere aktivieren kann, ist eine Zustandsüberwachung der Schale auf der Seite der Grafikanwendung unumgänglich.

Ein Sitzungsbildschirm soll flexibel aus der Wissensbasis heraus mit grafischen Elementen gestaltet werden können. Insbesondere sollen Schaltflächen, Texteingabefelder und Textausgaben, aber auch Grafiken frei platzierbar sein. Zur Sitzungszeit sollen so Benutzereingaben möglich werden. Darüberhinaus sollen Dialogfenster so konfigurierbar sein, daß der Benutzer auf etwas hingewiesen werden kann oder daß er Entscheidungen oder Aus-

wahlen treffen kann. Dies bedeutet, daß auch die Wissensbasis ereignisorientiert programmiert werden muß.

Die Wissensingenieurschnittstelle soll für die einzelnen Wissenskonstrukte spezialisierte Editoren besitzen. Die Trace-Ausgaben sollen in einem speziellen Werkzeug visualisierbar sein. Der Umfang der Ausgaben soll von dort aus wählbar sein.

Im Fehlerfall verzweigt der LISP-Interpreter in seinen eingebauten Debugger. Dieser ist von Produkt zu Produkt unterschiedlich ausgeprägt und zu bedienen [P12-P14]. Auch er soll über ein spezielles Fenster bedienbar sein. Bei Auftreten eines Fehlers soll dieses Fenster automatisch sichtbar werden.

Dazu muß ein hoher Aufwand in die Realisierung der Grafikanwendung gesteckt werden. Unter UNIX bietet die OSF/Motif Funktionenbibliothek ein ansprechendes Aussehen der Fenster und grafischen Elemente [U4]. Die geeignetste Programmiersprache zur Erstellung betriebssystemnaher Motifanwendungen sind sicherlich C und C++ [P4, P5]. Dies liegt in der gewachsenen Nähe von UNIX zu C begründet [U1, U2, U8]. Unter C stehen auch die Grafikbibliotheken des X11-Fenstersystems und der Erweiterungen von Motif zur Verfügung. Es lassen sich systemnah Interprozeßkommunikationskanäle handhaben und mit den X11-Bibliotheksfunktionen verbinden.

### **3.6 Umsetzung des vorgeschlagenen Lösungsansatzes unter UNIX-Betriebssystemen**

#### **3.6.1 Minimalversion der Gesamtanwendung**

Das oben definierte Gesamtziel soll in Teilschritten erreicht werden. Zuerst soll eine Grafikanwendung realisiert werden, die einen LISP-Interpreter startet, nachdem sie sich und noch zu bestimmende Interprozeßkommunikationskanäle initialisiert hat. Dieser LISP-Interpreter soll als erstes veranlaßt werden, eine LISP-Datei zu laden, die ihm ein für die Aufgabe notwendiges Mehr an Funktionalität verschafft.

Direkt nach der Abspaltung des LISP-Interpreters sind dessen Standardein- und Standardausgabe auf die Standardein- und Standardausgabegeräte der rufenden Grafikanwendung gerichtet. Dies ist der Normalfall. Auf ein in die Standardeingabe geschriebenes LISP-Kommando hin soll ein Dialogfenster aufgehen, das neben einem Text zwei Schaltflächen besitzt: Ja und Nein. Je nachdem, welche Fläche gedrückt wird, soll das Fenster abgebaut werden, und die Funktion mit wahr oder falsch zurückkehren. Der Vorgang

soll auf Standardein- und Standardausgaben verzichten. Mit einer derartigen Routine kann die Funktionalität Rückfrage beim Benutzer wie bei einer Terminalanwendung in den Programmfluß integriert werden. Die Anwendung läuft erst dann weiter, wenn der Benutzer reagiert und sich für eine der Varianten entschieden hat.

Da die Standardein- und Ausgabe nicht betroffen sein sollen, ist ein separates Kanalpaar bidirektional zu verwenden. Diese müssen benannte Pipelines sein, damit der LISP-Interpreter sie als Dateien öffnen kann. Um ihre Namen zu kennen, muß die Grafikanwendung ein Kommando in die Standardeingabe des LISP-Interpreters schreiben, was so nicht möglich ist.

Dies bedeutet, daß die Grafikanwendung zusätzlich eine Pseudoteletypeschnittstelle einrichten muß, im folgenden Grafikstrom genannt, um Standardein- und Ausgabe des LISP-Interpreters zu bedienen. Ferner muß sie die eigene Standardeingabe lesen, um sie in die Pseudoteletype durchreichen zu können. Ausgaben des Interpreters sind wiederum an die eigene Standardausgabe weiterzureichen. Dazu müssen die eigene Standardeingabe sowie die Pseudoteletype überwacht werden. Nach geeigneter Registrierung übernimmt dies der Eventhandler der X11-Funktionenbibliothek.

Nun kann dem LISP-Interpreter der initiale Befehl „Lade Datei XYZ“ gesandt werden. Je nach Produkt kann die Bootphase bis zum erstmaligen Erscheinen der Eingabeaufforderung unterschiedlich lang sein. Um zu verhindern, daß das abgesetzte Kommando mit Ausgaben des Interpreters kollidiert, müßte beispielsweise - während das Kommando gesendet wird - die ECHO-Einstellung der Pseudoteletype kurzfristig ausgeschaltet werden.

Die LISP-Datei kann die Namen der benannten Pipelines enthalten, sowie ein Kommando, das den Interpreter veranlaßt, diese auch seinerseits zu öffnen. Sicherheitshalber könnte er eine Bestätigungssequenz an die Grafikanwendung über diesen Kanal senden. Schließlich beinhaltet die Datei die Definition einer LISP-Funktion, die in Abstimmung mit dem beidseitig vereinbarten Protokoll eine Nachricht an die Grafikanwendung sendet und auf eine Antwort über den korrespondierenden Kanal wartet. Das Ergebnis wird eingelesen. Mit diesem Wert kehrt die Funktion zurück.

In der Grafikanwendung ist ein Parser zu realisieren, der die Pipeline abhört. Ihm sind die Token des Kommunikationsprotokolls bekannt. Beim Erkennen des Token für die Dialogerzeugung kann das Protokoll vorsehen, daß ein Text von der LISP-Seite anhängig ist, der bis zu einer Textendemarke andauert. Daraufhin wird der Dialog mit dem Text erzeugt. Weiterhin werden dem Dialog zwei Schaltflächen beigefügt, sowie Ereignisbehand-

lungsroutinen, die bei deren Betätigung erst den Dialog abbauen und dann die vereinbarten Rückgabewerte in die Pipeline schreiben.

Das so vereinbarte Protokoll ist stör anfällig, weil keine Empfangsbestätigungen vereinbart worden sind. Andererseits ist es synchron, weil LISP-seitig die Programmabarbeitung bis zum Erhalt der Antwort unterbrochen ist.

Die Grafikanwendung kann in der Zeit zwischen Erscheinen des Dialogs und dem Betätigen einer Schaltfläche durch den Benutzer beliebig auf Ereignisse reagieren. Damit ist das Teilziel grundsätzlich erreicht.

### **3.6.2 Ereignisorientierter Sitzungsablauf**

Sitzungen mit unter Babylon erstellten Expertensystemen werden ausschließlich über die Standardein- und Ausgabe abgewickelt, wie in Bild 2.10 verdeutlicht. Der Metaprozessor übernimmt die globale Kontrolle des Sitzungsablaufs. Zu diesem Zweck werden in der Wissensbasis die notwendigen Anweisungen zum Anstoßen von Regelwerken und dergleichen in den Block (`instructions ...`) geschrieben [S2, S3].

Sollen Sitzungen ereignisorientiert ablaufen, wie in Bild 2.11 angedeutet, sind unterschiedliche Arten von Benutzereingaben möglich. Eine den gestellten Anforderungen genügende Benutzerschnittstelle besteht aus einem Sitzungsfenster, auf dem in erster Linie mittels frei platzierter Druckknöpfe und Einträgen in Pulldownmenüs die Aktivierung von LISP-Kommandos ausgelöst wird. Dazu ist pro grafischem Element eine Reaktionsroutine registriert. Dabei handelt es sich um LISP-Kommandos, die in die Standardeingabe des LISP-Interpreters geschrieben werden. Diese wird damit zur Ereigniswarteschlange.

Nach Abarbeitung aller Einträge im Instruktionenblock ist eine Wissensbasis nicht notwendigerweise zu Ende. Der LISP-Interpreter ist nur wieder in seine Hauptschleife zurückgekehrt. Dieser Umstand kann dazu genutzt werden, daß nur noch die Initialisierung der Benutzerschnittstelle im Instruktionenbereich geschieht. Auf alle weiteren Ereignisse der Sitzung wird durch obiges Schreiben von LISP-Kommandos in die LISP-Standardeingabe reagiert.

### **3.6.3 Wechselnde Master-Slave-Beziehung**

Schon am Beispiel der Minimalversion aus Kapitel 3.6.1 wird deutlich, daß je nach Zustand der Gesamtanwendung die Grafikanwendung den LISP-Interpreter führen kann, dann wiederum ist es umgekehrt. Während der



Bootphase sowie bei Inaktivität des Systems ist die Grafikanwendung der Master, weil nur über sie Benutzeraktionen in das Gesamtsystem fließen können. Solange eine Sitzung läuft, geht die Kontrolle an den Interpreter über.

Die Editoren der Wissensingabeschnittstelle fragen beim LISP-Interpreter aktuelle Versionen einzelner Wissenskonstrukte ab, um sie bearbeiten zu können. Geänderte Versionen werden an den LISP-Interpreter zurückübertragen. Um einen Editor benutzen zu können, darf also keine Sitzung laufen, denn dann würden in die LISP-Standardeingabe formulierte Eingaben nicht bearbeitet werden. Erst bei Sitzungsende holt der Interpreter die Anfragen aus seinem Eingabepuffer und bearbeitet sie. Der Editor ist also während der Dauer einer Sitzung blockiert. Ansonsten kann die Antwort auf eine derartige Anfrage - angeführt von einem bestimmten Token - über den neu eingeführten Grafikstrom, ohne auf eine Rückantwort warten zu müssen, als Klartext übertragen werden.

Die Rückübermittlung des geänderten Wissenskonstrukts kann - eingekleidet in einen LISP-Befehl - wieder über die Standardeingabe des LISP-Interpreters erfolgen.

Dieses Konzept kann verallgemeinert werden. Solange der LISP-Interpreter inaktiv ist, also auf Befehlseingaben an seiner Standardeingabe wartet, können alle Aktionen, die vom Benutzer durch die Grafikanwendung ausgelöst werden, in entsprechende LISP-Kommandos umgesetzt und in die LISP-Standardeingabe geschrieben werden. Dies betrifft neben den Editoren alle Dateioperationen, sowie Änderungen des Umfangs des Traceprotokolls, aber auch aus dem Fenster der Erklärungskomponente heraus ausgelöste Anfragen.

#### **3.6.4 Interprozeßkommunikationskanäle**

In Kapitel 3.6.1 wurde hergeleitet, daß schon für eine minimale Anwendung eine Pseudoteletypeschnittstelle (für die Bedienung der Standard-ein- und Ausgabeströme) und 2 benannte Pipelines (für das Grafikprotokoll) benötigt werden. In Kapitel 3.4.2 wurden darüberhinaus ein bidirektionaler Sitzungsstrom, ein Ausgabestrom für Erklärungen, sowie 4 Ausgabeströme für Trace-Informationen benannt, die Babylon verwendet.

Dies bedeutet, daß diese Anwendung neben der Pseudoteletypeschnittstelle des xterm-Programms, von dem aus sie gestartet wurde, eine weitere Pseudoteletypeschnittstelle sowie 9 benannte Pipelines benötigt werden. In Anbetracht der Begrenztheit dieser Betriebsmittel sowie dem Umstand, daß

auch andere Anwendungen auf dem Rechner ablauffähig bleiben sollen, ist dieser Verbrauch zu groß.

Andererseits sind bei diesem Ansatz die geringstmöglichen Veränderungen der Quelltexte von Babylon durchzuführen.

Um den Verbrauch an Interprozeßkommunikationskanälen zu senken, muß zunächst die Kanalauslastung in den unterschiedlichen Betriebssystemen der Schale untersucht werden. Aus der Sicht des LISP-Interpreters existieren mit dem neu eingeführten bidirektionalen Grafikstrom 3 Eingabe- und 8 Ausgabekanäle.

- Ausgaben der Erklärungskomponente erfolgen in Klartext. Aufgrund der Eigenständigkeit dieses Mechanismus empfiehlt es sich, alle erklärenden Ausgaben der Inferenzmaschine in einem eigenen Ausgabefenster zusammengefaßt zu haben. Diese Textausgaben unterscheiden sich von der in Kapitel 3.6.1 vorgestellten Dialogbehandlung darin, daß LISP-seitig keine Antwort erwartet wird. Angeführt von einem eigenen Token kann die Textausgabe über den Grafikstrom versandt werden. Die Grafikanwendung verzweigt in eine Auswerteroutine, die die Textausgaben in dem entsprechenden Fenster der Erklärungskomponente erscheinen läßt.
- Die vier Trace-Ausgabeströme enthalten Klartextausgaben der einzelnen Fachinterpretier. Tracer sind Entwicklungshilfen, mit deren Hilfe der Ablauf der Wissensverarbeitung sichtbar gemacht werden kann. Der Wissensingenieur wird bei der Erstellung eines Expertensystems Tracer nutzen, wenn er beispielsweise auf der Suche nach Fehlern ist oder die Problemlösung beschleunigen will. Dazu ist eine chronologische Listung der Trace-Ausgaben oft geeigneter als eine thematische Trennung. Dementsprechend lassen sich die vier Trace-Ströme in einen einzigen unidirektionalen Kanal zusammenfassen, sofern sie um eine Kontextinformation ergänzt werden. Dies kann ein Kürzel vor der eigentlichen Nachricht sein. Pro Trace-Strom reicht es, ein weiteres Token in das Protokoll des Grafikstroms einzuführen.
- Mit der Entscheidung, die nicht mehr zeitgemäßen rein textstromorientierten Sitzungsabläufe nicht länger in der von Babylon gewohnten Form zu unterstützen, könnte auf den bidirektionalen Sitzungsstrom verzichtet werden. Standardein- und Ausgaben während der Sitzung wären durch geeignete grafische Elemente zu übernehmen. Die Ausgaben könnten in ein Textausgabefeld geschrieben werden. Die Eingaben dagegen könnten über Dialoge abgewickelt werden.

- Anpassungen der Standardein- und Ausgabe des LISP-Interpreters sind nicht möglich, weil die komplette Bootsequenz und die Debuggerbedienung im Fehlerfall nur hierüber abgewickelt werden können.
- Der Grafikstrom ist zur Handhabung grafischer Dialoge eingeführt worden. Die Vielfalt der Aufgaben hat hier von Anfang an eine Protokollfestlegung erfordert.

Es verbleiben also nur noch 2 bidirektionale Ströme.

### **3.6.5 Post-Mortem-Verhalten**

Die Anwendung soll nun um die Robustheitsforderung erweitert werden, daß beim unerwarteten Terminieren eines Programms die Kommunikationskanäle vom anderen Programm abgebaut werden, bevor es ebenfalls terminiert.

Wenn der LISP-Interpreter terminiert, teilt das Betriebssystem dies dem Elternprozeß Grafikanwendung über das Signal „Tod eines Kindes“ mit. Eine Signalbehandlungsroutine kann dann die Kanäle abbauen.

Wenn die Grafikanwendung terminiert, teilt dies der Pseudoteletype-Gerätetreiber dem LISP-Interpreter mit. Da seine Standardeingabe geschlossen wurde, terminiert dieser. Mit den in [P12, P13] beschriebenen Sprachmitteln ist eine Reaktion nicht möglich. Die Einträge der benannten Pipelines bleiben im Dateibaum stehen. Das Betriebssystem erklärt die verwendeten Geräte aber für frei.

### **3.6.6 Unterbrechungen**

Findet eine interaktive Sitzung mit einem LISP-Interpreter über ein Textterminal statt und wurde dabei ein Befehl angestoßen, dessen Abarbeitung zu lange dauert, kann die Bearbeitung unterbrochen werden. Dann speichert der Interpreter den aktuellen Kontext auf einem Stapel und verzweigt in seinen eingebauten interaktiven Debugger. Von dort aus kann wieder ein Befehl gestartet werden. Im Fehlerfall wird der Debugger erneut aufgerufen. Das Verlassen des Debuggers geschieht demgemäß rekursiv.

Um eine Unterbrechung einzuleiten, übermittelt das Textterminal ein Zeichen an den Gerätetreiber. Dieser veranlaßt den Scheduler, das Signal „Unterbrechungsanforderung von der Tastatur“ an den Interpreterprozeß zu senden. Bei Verwendung von Pseudoteletypes ist dies nachzurealisieren. In einem Menü der Grafikanwendung könnte es einen Menüpunkt Abbruch geben. Wird dieser angewählt, sendet die Grafikanwendung an ihren Kindprozeß, den LISP-Interpreter, das obige Signal.

### **3.6.7 Die Hauptschleife des LISP-Interpreters**

Ein Debugger ist in den meisten Interpreter-Sprachen vorgesehen, wenn auch mehr oder weniger rudimentär. Unter LISP stellt er einen definierten Wiedereintrittspunkt in die interaktive Betriebsart nach einem Fehler dar. Er stellt die Möglichkeit zur Verfügung, das Auslösen der Fehlerbedingung zurückzuverfolgen. Beispielsweise kann der Funktionsstack mit Aufrufparametern sowie eine große Zahl von Variableninhalten erfragt werden, was die Fehlersuche bei der Programmentwicklung unterstützt.

Bei interaktivem Betrieb eines LISP-Interpreters schreibt dieser zunächst eine Eingabeaufforderung und wartet danach auf die Eingabe eines kompletten LISP-Kommandos. Dieses kann sich über mehrere Zeilen erstrecken. Sobald es komplett eingelesen ist, wird es evaluiert und das Ergebnis dieser Operation wieder in die Standardausgabe geschrieben. Dieser Vorgang wiederholt sich ständig.

Diese Standardvorgehensweise kann beeinflusst werden. Zu diesem Zweck sind Eingriffsmöglichkeiten vorgesehen worden. In globalen Variablen können Verweise auf benutzerdefinierte Funktionen eingetragen werden, die einzelne Operationen ersetzen oder ergänzen sollen. Enthalten diese Variablen den Wert falsch oder existieren sie erst gar nicht, wird das Standardverhalten durchgeführt.

An erster Stelle ist die Möglichkeit zu nennen, die Standardfunktion für die Evaluierung eines LISP-Befehls zu ersetzen. Diese Möglichkeit kann dazu genutzt werden, um vor und nach dieser Standardfunktion Hilfsfunktionen auszuführen. Weiterhin ist es möglich, vor dem Einsprung in den Debugger eine eigene Funktionalität zu verankern, die über den weiteren Umgang mit der Fehlerbedingung entscheidet. Leider existiert keine Möglichkeit, zusätzlich Funktionalität an das Verlassen des Debuggers zu binden.

### **3.6.8 Virtuelle Kanäle**

Wenn auch die Zahl der Kanäle auf 4 begrenzt werden konnte, bleibt der Umstand, daß der bidirektionale Grafikstrom über benannte Pipelines abzuwickeln ist. In Kapitel 3.6.5 ist darauf hingewiesen worden, daß unter ungünstigen Umständen diese Pipelines nicht aus dem Dateisystem entfernt werden. Im Gegensatz dazu wird die Pseudoteletype der LISP-Standardein- und Ausgabe immer korrekt freigegeben. Weiterhin sind dem LISP-Interpreter immer zwei temporär erzeugte Namen zu nennen. Daher wird ein Weg vorgestellt, der es ermöglicht, die unterschiedlichen Aufgaben beider bidirektionaler Ströme zu vereinen.

Über den Grafikstrom werden bereits unterschiedliche Grafikoperationen beauftragt, indem LISP-seitig eine Protokolleröffnungssequenz mit Nachrichtentyperkennung (Token) gesandt wird, danach eine Nachricht und schließlich die Protokollendesequenz. In der Grafikanwendung sind Wächter registriert. Einer lenkt ankommende Daten der LISP-Standardausgabe in das Fehlerwerkzeug um. Der andere Wächter ordnet die Tokens des Grafikstroms entsprechenden Bearbeitungsrouitinen zu. Die Nachrichtentexte werden als Argumente diesen Routinen übergeben.

Der Grafikstrom kann genau dann auf die Standardausgabe gelegt werden, wenn das Grafikprotokoll als ununterbrechbar definiert wird. Während der Initialisierungsphase sendet LISP nur Standardausgaben. Sobald eine Grafik-Nachricht zu übertragen ist, wird die Protokolleröffnungssequenz geschrieben. Der modifizierte Wächter erkennt, daß eine Nachricht folgt und schreibt die empfangenen Zeichen nicht mehr in das Fehlerwerkzeug, sondern leitet bis zur Protokollendesequenz alle Zeichen der grafischen Auswertung zu.

Das Protokoll gerät außer Tritt, wenn zwischen Eröffnungs- und Endesequenz protokollfremde (Standard-)Ausgaben geschrieben werden.

Der Vorteil, zwei Kanäle zu besitzen, um in Richtung LISP-Interpreter zu senden, ist, daß ein neues LISP-Kommando bereits von der Grafikanwendung abgesetzt werden kann (beispielsweise „hole Regeldefinition“), während eine Sitzung noch läuft und ein Dialogfenster noch geöffnet ist. Die korrekte Abarbeitung der Eingaben ist gewährleistet. Werden beide Kanäle zusammengelegt, muß die Grafikanwendung erkennen, daß die Quittierung des Dialogs vor das bereits abgesetzte LISP-Kommando gereiht werden muß.

Durch Einbringen der Leistungserweiterung seitens der Grafikanwendung, die Eingabewarteschlange für LISP-Befehle umsordieren zu können, kann trotz einer Beschränkung auf einen einzigen LISP-seitigen Lesekanal die Funktionsfähigkeit aufrecht erhalten werden.

Diese Schlüsse lassen unter Beachtung der geforderten Protokolldisziplin eine Reduzierung auf den immer vorhandenen Standardein- und Ausgabekanal zu.

Die Interprozeßkommunikation kann schnell zum Flaschenhals werden, wenn viele Informationen ausgetauscht werden müssen. Daher sollte der Protokoll-Overhead klein gehalten werden. Wenn die Token der Kanaler-

kennung beispielsweise immer mit dem ASCII-Code **ESC** (Escape, Fluchtsymbol) beginnen, gefolgt von exakt drei Ziffern (0-9), dann ist das vierte Zeichen nach dem ESC immer das erste Zeichen der Nachricht. Somit stehen maximal **1000** unterschiedliche Nachrichtentypen oder **virtuelle** Interprozeßkommunikations**kanäle** zur Verfügung. Einige wenige Kanalnummern sind für besondere Aufgaben vorzubehalten. Per Konvention können dies die Kanalnummern 000 bis 009 sein. Darauf wird im folgenden Kapitel näher eingegangen. Als Protokollendesequenz soll das Umschalten auf einen dieser privilegierten Kanäle festgelegt werden.

Eine Nachricht könnte damit beispielsweise die folgende Form haben:

```
ESC 0 1 0 Z u g r i f f _ v e r w e i g e r t ! ESC 0 0 1
```

In der derzeitigen Implementierung ist der Kanal 010 gleichbedeutend mit der Aufforderung an die C-Seite, einen Informationsdialog mit dem Text „Zugriff verweigert!“ und einem Bestätigungs-Knopf zu erstellen (Bild 3.4) und LISP bis zur Quittierung durch den Benutzer warten zu lassen.



Bild 3.4: Ein Informationsdialog der um Grafikfähigkeit erweiterten Schale Babylon

Jede Grafikprotokollsequenz beginnt also mit einer Kanalnummer größer oder gleich 10 und endet mit einer Nummer kleiner 10. Die Protokolldisziplin verlangt, daß dazwischen keine protokollfremden Zeichen übertragen werden dürfen. Für jede verwendete Kanalnummer ist eine Behandlungsroutine zu registrieren, die die notwendigen Aktionen durchführt.

### 3.6.9 Zustandsmaschine als Beobachter des LISP-Interpreters

Der Forderung nach einer Fähigkeit der Grafikanwendung, die Reihenfolge der Eingaben für die LISP-Seite vertauschen zu können, wird dadurch Rechnung getragen, daß zunächst eine Funktion zur Verfügung steht,

die ein Kommando direkt absetzt. Sie heie *LISP\_sndReply()* und erwarte als einziges Argument einen kompletten, eventuell mehrzeiligen LISP-Ausdruck. Dieser wird - inklusive des abschlieenden ASCII-Zeichens Neue-Zeile (NL) - direkt in die Standardeingabe der LISP-Seite kopiert. Dieser Funktion berlagert existiere *LISP\_sndCmd()*, die ebenfalls einen kompletten LISP-Ausdruck erwartet. *LISP\_sndCmd* reiht diesen Ausdruck aber nur in eine Befehlswarteschlange (Queue) ein. Die Grafikanwendung pruft wiederholt, ob sich der LISP-Interpreter in einem Zustand befindet, in dem neue Kommandos eingegeben werden knnen. Dies sind die Eingabebereitschaft beim Warten in der Hauptschleife beziehungsweise im Debugger. Wenn LISP eingabebereit ist, wird gepruft, ob in dieser Befehlsqueue wenigstens eine Anweisung anliegt. Nach dem FIFO-Prinzip wird die lteste Eintragung mittels *LISP\_sndReply* in die LISP-Standardeingabe kopiert und der Eintrag in der FIFO gelscht.

Das im vorangegangenen Kapitel beschriebene Problem, da bei einem Dialogfenster gem Bild 3.4 eine Quittierung vor den eventuell bereits aufgelaufenen Kommandos in die Standardeingabe geschrieben werden mu, ist damit gelst. Whrend der LISP-Interpreter auf die Quittierung wartet, ist er nicht eingabebereit im oben beschriebenen Sinne. Damit werden die Kommandos zurckgehalten. Die Quittierung dagegen wird mittels der Funktion *LISP\_sndReply* direkt abgesetzt (daher auch der Name dieser Funktion).

Das Funktionieren dieses Konzepts setzt seitens der Grafikanwendung eine Kenntnis des aktuellen Zustands des LISP-Interpreters voraus. Eine geeignete Beobachtung ist zu ermglichen. Daher sind Zustandswechsel der Grafikanwendung zu signalisieren.

Der hufigste Fall ist, da sich der LISP-Interpreter in der Hauptschleife befindet und Eingaben erwartet. Das heit, das letzte Kommando wurde erfolgreich evaluiert und das Ergebnis in die Standardausgabe geschrieben. Danach wird der Text der Eingabeaufforderung ausgegeben. Dieser Zustand soll als eingabebereit oder **RDY** (ready) definiert werden. Sobald ein Kommando eingegeben und die Eingabe mit einem Zeilenumbruch-Zeichen (NL) abgeschlossen ist, wird diese Eingabe evaluiert. Bis zur erfolgreichen Ausgabe des Ergebnisses sei der LISP-Interpreter beschftigt oder **BSY** (busy).

Die virtuellen Kanle 000 bis 009 seien besonderen Aufgaben vorbehalten. Ein erweiterte Evaluierungsfunktion kontrolliert das Versenden der Synchronisierungsmittelungen an die Grafikanwendung. Dies geschieht

durch ein Ersetzen der Standardevaluierungsfunktion durch eine, die vor dem eigentlichen Evaluieren die Nachricht BSY und danach die Nachricht RDY versendet. Während der Evaluierung des Ausdrucks wird also auf den virtuellen Kanal 001 (BSY) umgeschaltet. Die Ergebnispräsentation erfolgt wieder über Kanal 000 (RDY). Beide Kanäle erwarten keine anschließenden Nachrichten. Die Ausgabe der Ergebnisse kann im Falle einer ausgedehnten Liste sehr lange dauern. Da die Grafikanwendung aber bereits den Zustand RDY detektiert hat, wird es eventuell schon ein neues Kommando absetzen. Dies behindert nicht das korrekte Arbeiten des LISP-Interpreters, kann aber zu vermischten Ausgaben im Fehlerwerkzeug führen.

Könnte man unter allen LISP-Produkten den Text der Eingabeaufforderung verändern, müßte die RDY-Sequenz nur in den Text der Eingabeaufforderung (Prompt) geschrieben werden. Damit ließe sich auch leicht eine Unterscheidung zwischen der Eingabebereitschaft der Hauptschleife und der des Debuggers erreichen.

Hier soll den eingeschränkten Möglichkeiten älterer LISP-Implementierungen Rechnung getragen werden. Der Prompt sei als implementierungsabhängig vorgegeben und nicht beeinflussbar anzunehmen. Der in der Grafikanwendung realisierte Beobachter muß bei Erhalt der RDY-Nachricht in kurzen Abständen prüfen, ob LISP-seitig noch Zeichen ausgegeben werden. Erst wenn der Ausdruck des Ergebnisses beendet und der Prompt geschrieben worden ist, wird auch durch den Beobachter die Eingabebereitschaft (RDY) angenommen.

Die nächste Grundsituation im Betrieb des LISP-Interpreters ist, daß beim Evaluieren einer Anweisung ein Fehler erkannt und die Abarbeitung unterbrochen worden ist. Dann wird implementationsabhängig eine Fehlermeldung geschrieben und in die **Debugger**-Betriebsart gewechselt. Vor diesem Wechsel kann eine eigene Fehlerbehandlung durchlaufen werden. Dort wird die Nachricht ESC 0 0 2 (**DBG**) an den Beobachter abgesetzt. Daraufhin macht die Grafikanwendung - falls noch nicht geschehen - das Fenster des Fehlerwerkzeugs sichtbar und löscht die Befehlswarteschlange. Falls dieses Ereignis während der Übertragung einer Grafiknachricht eingetreten ist, wird durch die Kanalnummer kleiner 10 die Nachrichtenübertragung korrekt beendet. Die Auswertung durch die für diese Grafiknachricht registrierte Auswertungsroutine wird aber verhindert und der empfangene Nachrichtentext gelöscht. Im Fehlerfall setzt der LISP-Interpreter die veränderte Funktionalität für die Evaluierung von LISP-Ausdrücken zurück. Daher wird vor dem Debuggereinsprung der spezialisierte Evaluator erneut installiert,



um auch während der Sitzung im Debugger den Beobachter geeignet informieren zu können. Schließlich wird in den Zustand RDY gewechselt, um Eingabebereitschaft zu signalisieren.

Als grafische Rückmeldung des aktuellen Zustands der LISP-Welt wird immer dann, wenn der Beobachter den Interpreter nicht als RDY einstuft, in sämtlichen Fenstern der Anwendung als Mauszeiger eine Uhr dargestellt. In allen übrigen Fällen wird die Standardform eines nach links oben deutenden Pfeils verwendet.

Die nächste Sonderbehandlung ist beim Verlassen der Debugger-Betriebsart zu berücksichtigen. Es kann dabei nicht davon ausgegangen werden, daß jede LISP-Implementierung das gleiche Kommando zum Verlassen des Debuggers verwendet. Wird dieses Kommando eingegeben, also der Fehlerbearbeitungsstapel um eine Ebene abgebaut, kann kein BSY gesendet werden, da die aktuelle Debugger-Hauptschleife gleichzeitig aufhört, zu existieren. Beim Verlassen des Debuggers wird der Zustand des LISP-Interpreters vor dem Fehlerfall wiederhergestellt, also beispielsweise auch die alte Evaluierungsfunktionalität. Die wiederhergestellte Hauptschleife wartet auf Eingaben, bevor sie ein erneutes BSY senden wird. Bei diesem Rücksprung wird demgemäß weder ein BSY noch ein RDY gesendet. Aus der Sichtweise des Beobachters ist noch nicht mit der Bearbeitung des letzten Kommandos, das ja den Rücksprung aus dem Debugger eingeleitet hat, begonnen worden. Deshalb blockiert der Beobachter die Weiterleitung weiterer Kommandos. Eine **Verklemmung** ist eingetreten.

Daher wird jedesmal, wenn versucht wird, ein Kommando auf die LISP-Seite zu senden, zusätzlich eine Flag-Variable gesetzt. Nach einer gewissen Zeit, in der diese Flag-Variable gesetzt ist, ist mit mindestens einem Kanalwechsel (nämlich dem nach BSY) zu rechnen. Jeder Kanalwechsel setzt dieses Flag zurück. Unterbleibt dieser Kanalwechsel, ist davon auszugehen, daß der Sonderfall „Rückkehr aus dem Debugger“ vorliegt. Es genügt, die Zustandsmaschine explizit in den Zustand RDY zu überführen und das Flag zurückzusetzen.

**Grafiknachrichten** werden von der LISP-Seite nur gesandt, nachdem der Zustand BSY erreicht worden ist. Aufgrund der vorausgesetzten Disziplin werden sie immer korrekt mit einem Zurückschalten nach BSY beendet. Ein Sonderfall ist auch hier zu beachten. Die derzeit beschriebenen Mechanismen verhindern eine erfolgreiche Verwendung der LISP-Funktion *read*. *Read* wird in den Quelltexten von Babylon wiederholt verwendet, um einen

LISP-Ausdruck aus der Standardeingabe zu lesen. Das Umschalten nach BSY vor der Ausführung von *read* verhindert, daß die Eingabe des Benutzers über das Texteingabefenster des Fehlerwerkzeugs an die Funktion *read* weitergeleitet wird. Die Grafikanwendung geht davon aus, daß es sich dabei um ein weiteres Kommando handelt, das zunächst in die Befehlswarteschlange eingereiht wird. Daher muß anstelle von *read* eine Ersatzfunktion verwendet werden, die vor dem eigentlichen Aufruf von *read* auf den Sonderkanal 003 umschaltet. Diese Nachricht veranlaßt die Grafikanwendung, die nächste Eingabe statt mit *LISP\_sndCmd*, mit *LISP\_sndReply* weiterzuleiten. Abschließend wird ein BSY gesendet.

Hinsichtlich der hierbei notwendigen LISP-Quelltextanpassung sei angemerkt, daß jedes Vorkommen von *read* durch eine andere Funktion zu ersetzen wäre. Es ist aber auch möglich, den bereits modifizierten Evaluator-Aufruf weiter anzupassen, sodaß er vor jeder Evaluation - also auch bei Rekursionen - prüft, ob ein *read* zu evaluieren sei. Er kann die Evaluation mit Beginn- und Endesequenzen umfassen.

Das Anfordern von **Dialogfenstern** durch die LISP-Seite ist bereits beschrieben worden. Aufgrund der verwendeten Pseudoteletypes werden auch die Echos der Antworttexte auf die LISP-Standardausgabe geschrieben. Um das Textausgabefenster des Fehlerwerkzeugs, das alle LISP-seitigen Ausgaben protokolliert, nicht zu überfluten, wird die Grafiknachricht, die den Aufbau des Dialogfensters bewirkt, nicht mit BSY beendet, sondern mit einem Wechsel in den Sonderkanal 004. Erst nach Erhalt der Quittung wird nach BSY zurückgeschaltet. In der Zwischenzeit wird die Ausgabe in das Fehlerwerkzeug durch den Beobachter unterdrückt.

Die **Bootphase** kann gegenüber dem in Kapitel 3.6.1 vorgetragenen Konzept ergänzt werden. Im Falle einer Weiterentwicklung der Expertensystemschale muß nach der Abspaltung des LISP-Interpreters zunächst das Laden einer LISP-Datei initiiert werden, die wiederum das Laden der Erweiterungsmodule in den Interpreter bewirkt. Im regulären Betrieb sind diese Erweiterungen bereits Bestandteil des aktivierten LISP-Interpreters. In jedem Fall wird als Signal für die Beendigung dieses Teils der Bootphase die Sequenz ESC 0 0 9 (**EOI**, End of Initialisation) als Synchronisierungsnachricht an die Grafikanwendung gesendet.

Babylon ist in der Lage, mehrere Wissensbasen gleichzeitig im Speicher zu halten. Der Sonderkanal 008 soll vorgesehen werden, um die aktuellen Zustände sämtlicher aktuell in der Schale geladenen Wissensbasen der Grafikanwendung bekanntzugeben. So weiß diese zu jedem Zeitpunkt, wie-



men. Vielmehr zeigen die fetten Pfeile die Zustandsänderungen des LISP-Interpreters ab der Bootphase auf. Die rechteckigen Felder symbolisieren die einzelnen Zustände mit den zugeordneten Nummern der virtuellen Kanäle, respektive den Token, die eine Zustandsänderung auslösen. Die dünnen Pfeile verdeutlichen, daß prinzipiell von jedem Zustand aus ein Einsprung in den Debugger möglich ist. Die Rauten symbolisieren zusätzliche spezielle Rechenoperationen, die mit dem Zustandswechsel einhergehen. Die Rauten sowie der fette gestrichelte Pfeil behandeln nur den Sonderfall Rückkehr aus dem Debugger.

Zusammenfassend läßt sich bemerken, daß für eine Erweiterung des vorhandenen Leistungsumfangs der Schale Babylon nur geringfügige Änderungen der vorhandenen LISP-Quelltexte vorgenommen werden müssen. Das Prinzip der virtuellen Kanäle ist - nachdem es einmal realisiert worden ist - flexibel und leicht erweiterbar. Auch Erweiterungen des Leistungsumfangs der Schale wie die Realisierung einer Wissensingenieurschnittstelle werden daraufhin mit geringem Aufwand in beiden Programmen möglich.

### **3.7 Beispiel für die Verallgemeinerbarkeit des nebenläufigen Ansatzes**

Dieser nebenläufige Ansatz ist verallgemeinerbar. Beispielsweise ist am Lehrstuhl für elektrische Maschinen und Antriebe eine Simulationssoftware entwickelt worden, die die berechneten Kennlinien nur auf Plottern und Grafikbildschirmen von Singletaskingbetriebssystemen ausgeben konnte [G1]. Das grafische Ausgabegerät stand der Anwendung also immer exklusiv zur Verfügung. Um auch auf zeitgemäß ausgestatteten Arbeitsplatzrechnern genutzt werden zu können, sollte die Grafikfunktionenbibliothek dieser Software auf den Einsatz in Mehrfenstertechnik umgestellt werden. Immer dann, wenn in der Anwendungslogik einer Simulationsaufgabe die Öffnung des grafischen Ausgabegeräts gefordert wird, soll auf dem Display ein Fenster geöffnet werden, in das gezeichnet werden kann. Dieses Fenster besteht, bis das Simulationsprogramm das Ausgabegerät wieder schließt. Anders als bei verfügbaren Grafikpräsentationsanwendungen wie [G3] soll der Fortschritt der Simulationsrechnung unmittelbar verfolgt werden können, die Darstellung der Kurven also bereits zur Simulationszeit erfolgen. Dazu muß sie auch in der Lage sein, nach Verdeckungsaufhebungen den Bildschirminhalt wiederherzustellen. Bereits gezeichnete Linien dürfen hier nicht verlorengelassen. Für interaktive Anwendungen müssen darüber hinaus Mausereignisse an die Anwendungslogik weitergeleitet werden können. So können beispielsweise Kennlinienverläufe grafisch modelliert werden. Dies geschah

zunächst durch Setzen einer Ausnahmebehandlungsroutine für Mausereignisse, die sämtliche Zeigerbewegungen und Zustandsänderungen der Drucktasten auf der Maus feststellte.

Um nicht in die Ablauflogik der Simulationsrechnung eingreifen zu müssen, ist auch hier die Entscheidung zugunsten einer nebenläufigen Aufgabenteilung gefallen. Mit der Öffnung des Ausgabegeräts wird ein nebenläufiger Grafikserver abgespalten. Aufgrund der mit hoher Geschwindigkeit auflaufenden großen Datenmengen wird in diesem Fall als Mittel der Interprozeßkommunikation ein Ringpuffer in einem gemeinsam genutzten Speicherbereich (sharable memory) eingerichtet. Ein einfaches Protokoll beschreibt die Zeichenoperationen, die durchzuführen sind. Im Grafikserver werden die Bildinhalte unsichtbar im Hintergrund gepflegt und nach Bedarf Kopien von Teilflächen in den sichtbaren Teil des Fensters kopiert. Dies findet beim Neuzeichnen, aber auch bei Verdeckungsaufhebungen durch Änderungen im Fensterstapel statt.

Der Grafikserver erhält vom unterliegenden Fensterbetriebssystem sämtliche Informationen vom Zustand der Maus, sofern sich der Mauszeiger im sichtbaren Bereich der vom Server kontrollierten Fenster befindet. Diese Informationen werden ebenfalls in dem gemeinsam genutzten Speicherbereich abgelegt. Die Anwendungslogik des Simulationsprogramms wird von jeder Änderung der Mausereignisdaten über ein bestimmtes Signal informiert. Die Signalbehandlungsroutine prüft, ob eine anwendungsspezifische Auswerteroutine registriert worden ist, und ruft diese gegebenenfalls auf.

Dieses Beispiel zeigt, daß durch den nebenläufigen Ansatz komplexe vorhandene Ablauflogik unverändert bleiben konnte. Die hinzugekommene Funktionalität ist in den grafischen Bibliotheksfunktionen untergebracht worden. Deren Aufrufverhalten hat sich dadurch nicht verändert. Aus Sicht der Ablauflogik verhält sich das grafische Ausgabegerät Fenster nicht anders als ein Plotter. Diese Sicht läßt sich bis zur Befüllung des Ringpuffers aufrechterhalten. Auf der anderen Seite konnte das Problem der Interaktivität gelöst werden, indem eine von der Hardware ausgelöste Unterbrechung durch das Synchronisationsmittel Signal nachgebildet worden ist. Aufgrund der geradlinigen Anforderungen an die Kommunikation sind hier weder eine aufwendige Beobachtung der Partneranwendung noch eine virtuelle Kanalvervielfachung notwendig.

### 3.8 Zusammenfassung

In Kapitel 3.6 ist ein Konzept vorgestellt worden, mit dem die Schale Babylon ohne umfangreiche Modifikationen der Quelltexte um grafische Dialogkomponenten erweitert werden kann.

Dieses Konzept ist in Form der hybriden grafischen Expertensystem-schale EMA-XPS am Lehrstuhl für elektrische Maschinen und Antriebe der Universität Wuppertal umgesetzt worden und hat sich dabei als robust und stabil erwiesen. EMA-XPS ist auf unterschiedliche UNIX-Derivate portiert und mit mehreren LISP-Interpretern getestet worden.

Geringfügige Probleme traten mitunter bei der Grafikanwendung im Bereich der Einrichtung der Pseudoteletypes auf, da die Namenskonventionen und Pfadangaben von System zu System schwanken. Bei den LISP-Interpretern liegen die Probleme in der Installation der Sonderbehandlungsroutinen zur Evaluierung und dem Debuggereinsprung.

Mit EMA-XPS konnte damit eine leistungsfähige Schale zur Verfügung gestellt werden, die

- im UNIX-Umfeld leicht portabel ist,
- eine hybride Inferenzmaschine besitzt,
- grafikfähig ist und
- unter wirtschaftlichen Bedingungen unter einem Betriebssystem lauffähig ist, das die notwendigen Ressourcen bietet, um eine Auslegung von Schaltnetzteiltransformatoren durchzuführen.

Das vorgetragene Konzept schont die Betriebsmittel, indem es durch Einführung eines Beobachters innerhalb der Grafikanwendung den Zustand des LISP-Interpreters überwacht. Basierend auf nur einem bidirektionalen Datenstrom können LISP-seitige Ausgaben kategorisiert versandt werden, indem das Betriebsmittel Übertragungskanal virtuell vervielfacht wird. In der umgekehrten Richtung ermöglichen zwei Arten der Versorgung der LISP-seitigen Standardeingabe einen kontextbezogenen Informationszufluß, also ebenfalls eine virtuelle Vervielfachung des Eingabekanals.

## **4. Strukturierung und Implementierung des akquirierten Fachwissens**

Wie bereits in Kapitel 1 verdeutlicht werden konnte, ist die Auslegung von Mittelfrequenztransformatoren in primärgetakteten Schaltnetzteilen ein mühseliges Unterfangen. Eine Unterstützung des Entwicklungsingenieurs bei der Auslegung durch einen Assistenten verkürzt die Entwicklungszeit und erleichtert so die Annäherung an das angestrebte Optimum. Dabei sind unterschiedlichste Abwägungen bis in den Schaltnetzteilfertigungsprozeß sowie die Gesamtanlage, zu der das Netzteil gehört, aber auch Wirtschaftlichkeitsbetrachtungen geeignet, als Optimierungsparameter zu dienen. Im Rahmen dieser Arbeit soll eine leicht erweiterbare, auf die technischen Parameter ausgerichtete Vorgehensweise erarbeitet werden.

### **4.1 Betrachtung der Vorgehensweisen bei der Auslegung**

Dem Entwicklungsingenieur werden die Rahmenbedingungen mitgegeben, unter deren Berücksichtigung er ein neues Schaltnetzteil realisieren soll. Diese Rahmenbedingungen sind die Eingabegrößen einer Optimierungsaufgabe.

- Für den laufenden Betrieb des Netzteils werden dies an vorderster Stelle Verlustarmut und Betriebssicherheit sein.
- Für den Herstellungsprozeß bedeutet dies, daß aufgrund fertigungstechnischer Rahmenbedingungen bevorzugt ausgewählte Materialien zu verwenden sind. Dies können bestimmte Produktlinien von Magnetmaterialianbietern sein, oder auch nur Teilbereiche davon. Es ist auch möglich, daß die Menge der Halbleiterprodukte beschränkt wird.
- Aus technischer Sicht sind die Anforderungen einerseits durch die Beschaffenheit des speisenden Netzes samt zulässiger Netzurückwirkung bestimmt. Andererseits sind lastseitig maximale Leistungsabgabe, Spannungsfestigkeit und Restwelligkeit entscheidende Parameter.
- Weiterhin spielt das Umfeld eine Rolle, in die das Netzteil eingebettet wird. Hier sind die thermische und elektromagnetische Wechselwirkung mit der Umgebung, aber auch Bauhöhenbeschränkungen, beispielsweise in Einschubsystemen, vorstellbar. Gegebenenfalls müssen Abschirmbleche und Lüfter eingesetzt werden.

Wie bereits in Kapitel 1 aufgezeigt, sind diese Vorgaben in ihrer Zielsetzung mitunter zueinander gegenläufig.

Die Entscheidung, ein Netzteil als primärgetaktetes Schaltnetzteil zu realisieren, impliziert bereits eine Zielsetzung hin zu kostengünstiger Fertigung und geringen Verlusten, aber auch hin zu geringem Gewicht und Bauvolumen des Netzteils.

Diese Überlegungen führen zunächst dazu, daß eine bestimmte Menge an verfügbaren Kernen gebildet wird, die als Kandidaten für das zu realisierende Netzteil in Frage kommen. Weiterhin kann die Menge der verwendbaren Schaltnetzteiltopologien bereits einzuschränken sein. Für jeden Kern aus dieser **Startmenge** müssen pro zu testender Topologie ein geeigneter Auslegungsalgorithmus einschließlich aller von diesem Algorithmus benötigten Parameter vorhanden sein.

Bei **systematischer** Vorgehensweise sind nun für alle Kerne und Topologien die jeweiligen Auslegungsalgorithmen abzarbeiten. Es entsteht eine **Ergebnismenge** aus Kern-Topologie-Kombinationen. Wenn  $n$  Kerne in  $m$  Topologien zu untersuchen sind, hat die Ergebnismenge  $n \cdot m$  Elemente.

Unter Zuhilfenahme einer vorab festgelegten Bewertungsfunktion kann für jedes Element der Ergebnismenge aufgrund der für diese Kombination ermittelten Ergebniswerte eine Einordnung in einer Skala vorgenommen werden. Diese kann beispielsweise Werte von 0 für ungeeignet bis 100% für sehr geeignet definieren. So entsteht eine Bestenliste.

Über die auf diese Skala abgebildeten Einzelergebnisse der Bewertungsfunktion können beispielsweise Schwellwerte, Mittelwert und Standardabweichung gebildet werden. Mit diesen Größen lassen sich qualifizierte Aussagen über die gewählte Startmenge der Kerne und die in Frage kommenden Topologien ableiten. Im Beispiel wäre für Mittelwert und Standardabweichung nahe 0 die Aussage abzuleiten, daß mit den zur Verfügung gestellten Kernen das gewünschte Schaltnetzteil prinzipiell nur unbefriedigend realisierbar ist. Entweder ist die Startmenge anders zu wählen oder die Anforderungen an das zu realisierende Netzteil einschließlich der Topologievorgaben sind zurückzunehmen. Um bessere Ergebnisse erzielen zu können, ist der Auslegungsvorgang unter angepaßten Ausgangsbedingungen zu wiederholen.

Ein **erfahrener** Schaltnetzteilentwickler wird von dem beschriebenen Verfahren abweichen. In grundsätzlicher Kenntnis des zu erwartenden Ergebnisses verkleinert er bereits die Startmenge. Er untersucht nur erfolgversprechende Kern-Topologie-Paarungen. Um sich die Arbeit weiter zu vereinfachen, wird er zunächst versuchen, die Startmenge weiter einzuschränken. Daher prüft er zunächst einfache Ausschlußkriterien. Aufgrund der



Luftspaltproblematik sind beispielsweise nicht alle Kerne für sämtliche Topologien einsetzbar. Die zeitaufwendigen Auslegungsalgorithmen müssen dann nur noch auf die Restmenge angewandt werden.

Es gilt nun festzustellen, was für eine Art von Assistenz sich ein erfahrener Entwickler wünschen könnte.

- An erster Stelle ist sicherlich die **automatisierte Abarbeitung der Auslegungsalgorithmen** zu nennen, da sie für ihn reine Fleißarbeit darstellt. Bei korrekt vorgegebenen Eingangsdaten können dem jeweiligen Algorithmus ohne Unsicherheit korrekte Ergebnisse entnommen werden. Die anschließende Bewertung dagegen läßt wieder Interpretationen zu.
- Wenn eine **Bewertungsfunktion** leicht konfektionierbar implementiert wäre, könnte er diese problembezogen parametrisieren und so die Aufstellung der Bestenliste teilautomatisieren.
- Weiterhin würde eine bereits gepflegte **Parameterdatenbasis** mit zu jeder Kern-Topologie-Kombination zugehörigen **Auslegungsalgorithmen** wiederholte Eingaben erübrigen. Andererseits müßte diese Datenbasis leicht erweiterbar sein, falls Kerne hinzugefügt werden sollen. Ebenfalls sollten neue Auslegungsalgorithmen leicht implementierbar sein.

Diese Aufgaben könnten mit unterschiedlichen Softwarewerkzeugen realisiert werden. Unter Bezug auf die Auslassungen in den Kapiteln 2.1 und 2.2 würden sehr überschaubare Datenmengen den Einsatz einer Datenbank nicht notwendigerweise rechtfertigen. Die Implementierung der Auslegungsalgorithmen kann in den meisten Programmiersprachen vorgenommen werden. Dies gilt auch für die Realisierung einer parametrisierbaren Bewertungsfunktion. Wenn diese Parametrisierung interaktiv zur Sitzungslaufzeit grafisch unterstützt vorgenommen werden soll oder wenn Ergebnisse grafisch aufbereitet auszugeben sind, ist ein ereignisorientiert arbeitendes grafisches Softwarewerkzeug von Nöten. Da die Abarbeitung der Auslegungsalgorithmen gegebenenfalls längere Zeit in Anspruch nehmen kann, sind darüberhinaus die Überlegungen aus Kapitel 3 zu berücksichtigen, die eine Nebenläufigkeit von grafischer Ereignisabarbeitung und Auslegungsalgorithmen ermöglichen.

Ansonsten hat der Entwickler sämtliche Entscheidungen in Bezug auf den Auslegungsvorgang selbst zu treffen. Daher sollen im weiteren die Wünsche eines weniger erfahrenen Schaltnetzteilentwicklers hinzugenommen werden.

Ein gelegentlicher Schaltnetzteilentwickler kann nicht auf das Erfahrungswissen des erfahrenen Entwicklers zurückgreifen. Er ist gezwungen,

systematisch vorzugehen. Hier ist der Einsatz eines Expertensystems geboten, denn diese **Heuristiken** lassen sich darin am komfortabelsten umsetzen. Weiterhin kann der Benutzer über die erweiterten Mittel der Erklärungs-komponente den vom System eingeschlagenen Lösungsweg nachvollziehen, um daraus für sich selbst Schlüsse zu ziehen.

Andererseits wird auch ein erfahrener Entwickler ein derartiges wissensbasiertes System bevorzugen, insbesondere, wenn es ihm die Möglichkeit bereitstellt, gegebenenfalls im Dialog aus vage formulierten Beschreibungen der Rahmenbedingungen für die Schaltzerteilauslegung exakte Vorgaben für die Auslegungsalgorithmen abzuleiten. Beispielsweise kann die Vorgabe, daß die Restwelligkeit der Ausgangsspannung weniger wichtig ist, die Wahl von Sperrwandlertopologien begünstigen. Auch kann er die Startmenge aus der Grundgesamtheit des Kerndatenbestands von dem System zusammengestellt lassen. Diese Auswahl wird er gegebenenfalls noch modifizieren, kann so aber mit wenig eigenem Aufwand bereits die Auslegungsalgorithmik anstoßen. Nur bei allgemein unbefriedigenden Ergebnissen muß er, wie oben beschrieben, seine Erfahrung bei der Modifizierung der Startmenge oder der Rahmenbedingungen einbringen. Beispielsweise wird er auf Magnetmaterialien ausweichen, die er sonst aus unterschiedlichsten Gründen nicht verwendet hätte.

## **4.2 Anforderungen an den Sitzungsablauf**

In dem gewählten Anwendungsschwerpunkt des zu erstellenden Expertensystems kommt als Problemlösungstyp die Konstruktion in Frage. Der Transformator wird aus Komponenten zusammengesetzt. Dabei muß zunächst eine Planungssequenz durchlaufen werden, in der der Anwender nach seinen Anforderungen an den gesuchten Transformator befragt wird. Der Anwender muß dabei beispielsweise auch die Möglichkeit haben, die Anzahl der zur Verfügung stehenden Materialien einzuschränken. Es ergeben sich in dem Anwendungsschwerpunkt zunächst zwei grundlegend verschiedene Vorgehensweisen der Problemlösung für den Anwender.

- 1) Der Anwender gibt dem System ein Anforderungsprofil vor, in dem Ausgangsleistung, Störstrahlung, Gewicht, Größe, und dergleichen vorge-schrieben werden. Das Expertensystem wählt daraufhin Kern und Be-wicklung aus.
- 2) Der Anwender gibt Kern und Wickelmaterial vor, und das Expertensy-tem wertet aus, welche Leistung mit einem daraus zu erstellenden Transformator maximal übertragen werden kann.

Die erste Fragestellung bedingt eine vorwärtsverkettete Regelverkettung. Dabei sollte heuristisches Wissen genutzt werden, um die Regelarbeitung zeitlich begrenzen zu können. Der zweitgenannte Fall ist zielgerichtet und damit gut geeignet, mittels rückwärtsverketteter Regelverarbeitung gelöst zu werden. Der erstgenannte Fall entspricht der hier gestellten Aufgabenstellung und soll im folgenden vertieft werden.

Gemäß den in Kapitel 4.1 behandelten Zielgruppen bieten sich zwei Vorgehensweisen für eine Sitzung mit dem Expertensystem an:

- 1) Eine bestimmte Gleichspannung ist bei minimal möglichem Volumen, Masse oder Streufeld galvanisch entkoppelt bereitzustellen. Welche Schaltnetzteiltopologie ist geeignet und wie ist der Mittelfrequenztransformator anzufertigen? Hierbei wird zunächst nur das Umfeld, in das der Transformator eingebettet ist, insbesondere die Schaltnetzteiltopologie, bestimmt. Weiterhin werden die Rahmendaten für die eigentliche Transformatorberechnung, wie die Ermittlung der primärseitigen Gleichrichtspannung  $U_d$ , bereitgestellt.
- 2) Die Netzteiltopologie, die sich durch die gewählten Leistungshalbleiter ergebenden Beschränkungen und der Wert der primärseitigen Gleichrichtspannung  $U_d$  seien bekannt. Ein bestimmte Leistungsabgabe werde bei minimal möglicher Baugröße, Masse oder Streufeld benötigt. Welche Kerne sind geeignet und wie lauten die Wickelvorschriften?

Es zeigt sich, daß die erste Vorgehensweise der zweiten vorgeschaltet gesehen werden muß. Die Zielsetzungen können von einander unabhängig betrachtet werden. Daher empfiehlt sich eine Trennung der Gesamtheit der zu erstellenden Regeln in **zwei Regelmengen**.

Dies hat unmittelbar Einfluß auf die Gestaltung des Sitzungsablaufs. Jede Sitzung beginnt grundsätzlich mit dem Bildaufbau des Sitzungsfensters. In interaktiver Betriebsart wählt der Benutzer zunächst, ob er eine Gesamtauslegung wünscht, also die Schritte 1 und 2, oder ob er direkt mit Schritt 2 beginnen will. Zu Beginn jedes Schrittes wird dem Benutzer die Gelegenheit gegeben, seine Vorgaben für die Sitzung interaktiv einzugeben. Hier wäre eine Halb- oder Vollautomatik unter Zuhilfenahme von Eingabedateien vorstellbar (Batchbetrieb). Bei Halbautomatik würden die Eingabe wie Vorbelegungswerte in die interaktive Sitzung einfließen, bei Vollautomatik würde unter Zuhilfenahme dieser Werte auf die Interaktivität verzichtet werden und das jeweilige Regelwerk direkt angestoßen werden. Bei interaktivem Betrieb sollte dem Benutzer Assistenz durch Eingabehilfen in Form von Hilfetexten und Bildern angeboten werden.

Nach Beendigung der Regelverarbeitung sollten die ermittelten Ergebnisse ebenfalls in Form von Texten (beispielsweise die Wickelvorschriften) und Bildern (von Schaltplänen oder Kernbauformen) bereitgestellt werden. Darüberhinaus werden als Ergebnis der ersten Regelmenge Vorbelegungswerte für die Eingabemaske des zweiten Sitzungsschritts erzeugt.

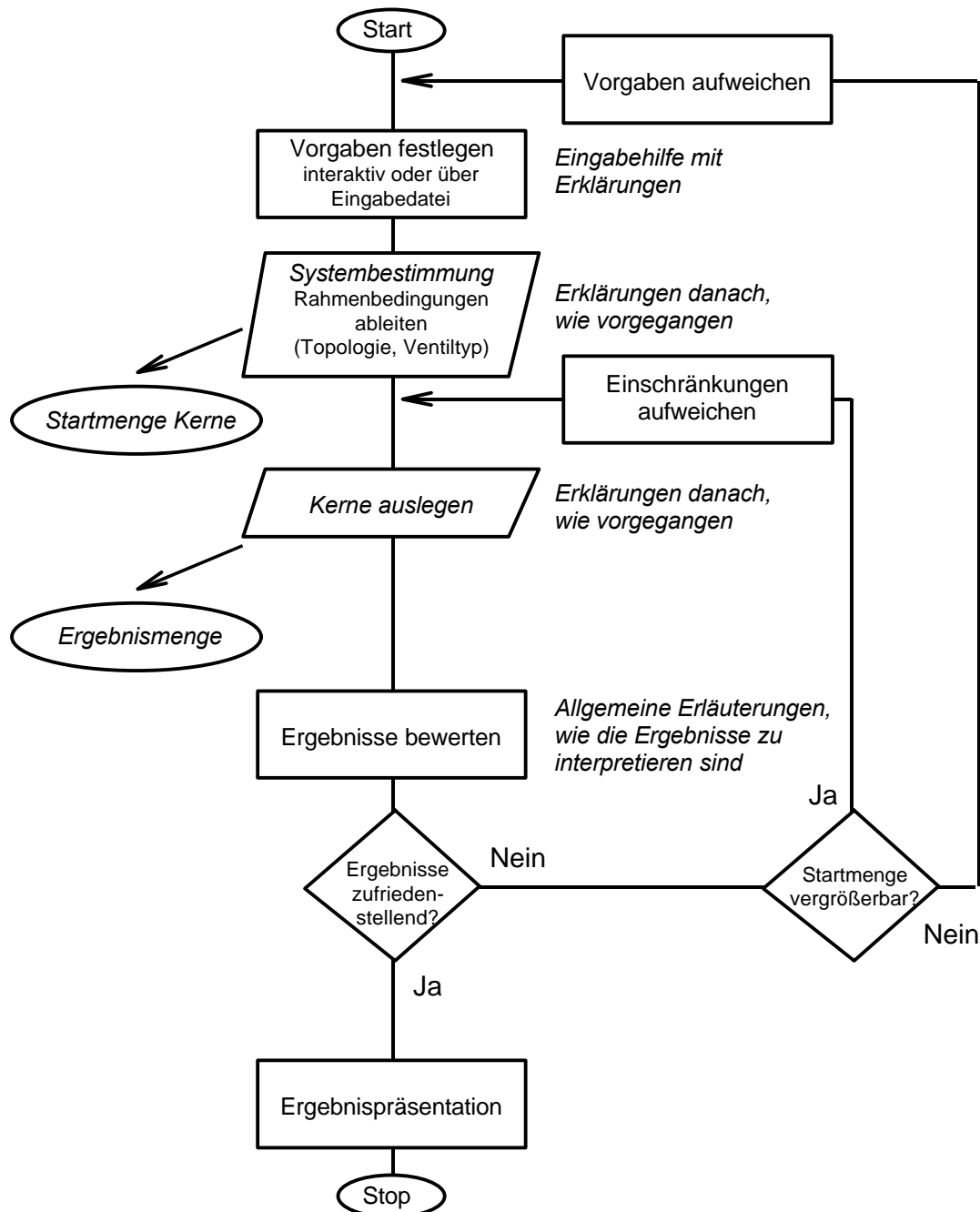


Bild 4.1: Flußlaufplan eines Sitzungsablaufs

Der Benutzer sollte für ihn interessante Informationen über den Fortschritt der Regelverarbeitung protokolliert und auf dem Sitzungsbildschirm angezeigt bekommen. Bei Rückfragen des Systems im Zusammenhang mit der Regelverarbeitung sollten die erscheinenden Dialoge ebenfalls Hilfestellung und Hintergrundinformationen anbieten.

Bei der abschließenden Bewertung der ermittelten Ergebnisse sollte deren Bewertung nachvollzogen werden können. Im Falle grundsätzlich unbefriedigender Ergebnisse wäre Hilfestellung bei der Änderung der Ausgangsdaten wünschenswert. Dann ist der Auslegungsprozeß zu wiederholen. Die geänderten Vorgaben stellen dabei den Zuwachs an Wissen dar.

Die in diesem Kapitel erörterten Betrachtungen spiegeln sich in dem Flußlaufplan (Bild 4.1) wieder.

### 4.3 Wissensstrukturierung

Aus der Sicht eines Wissensingenieurs, dessen Aufgabe die Erstellung dieses Expertensystems ist, stellt sich das in Kapitel 1 zusammengetragene Fachwissen als das Ergebnis einer **Wissensakquisition** dar. Der Wissensingenieur muß nicht notwendigerweise auch Kenntnis davon haben, wie Schaltnetzteile auszulegen sind. Er kann diese Informationen auch durch Befragung eines Schaltnetzteilentwicklers als Experten gesammelt haben.

Der zweite Arbeitsgang, den der Wissensingenieur praktiziert, ist die **Wissensstrukturierung**. Er legt das Ziel der Anwendung fest. Danach wählt er typischerweise eine Expertensystemschale aus, prüft die Mächtigkeit der angebotenen Wissensrepräsentationsformen und ordnet die Einzelinformationen aus der Stoffsammlung diesen Mechanismen zu. Gegebenenfalls ist dieser Vorgang für unterschiedliche Schalen beziehungsweise Entwicklungsumgebungen zu wiederholen. Schließlich wird in Kenntnis der Wechselwirkungsmöglichkeiten der einzelnen Wissensrepräsentationsformen der Weg der Lösungsfindung konzipiert.

Danach kann mit der **Wissensimplementierung**, also der tatsächlichen Eingabe der Konstrukte über die Wissenserwerbskomponente (typischerweise spezialisierte Editoren), begonnen werden. Die Wissensbasis wird befüllt. In zeitgemäßen Anwendungen kann diese Wissensbasis als eine Datei abgelegt und wieder eingelesen werden. Diese Trennung der Schale von der Wissensbasis ermöglicht eine flexiblere Nutzung der Software.

Nach einer Test- und Abnahmephase kann das Expertensystem zur Benutzung freigegeben werden. Zu Testzwecken ist es vorteilhaft, wenn Trace- und Fehlerwerkzeuge bereitstehen. Die Informationen der Erklärungskom-

ponente beispielsweise über den Zustand der Regelverarbeitung helfen hier ebenfalls weiter.

Kommt neues Wissen hinzu, das in das System zu integrieren ist, in diesem Fall werden dies bevorzugt neue Kerne oder präzisere Auslegungsalgorithmen sein, so ist der benannte Weg zu wiederholen.

Schwerpunkt dieses Kapitels ist eine zielgerichtete Strukturierung und Bewertung der in Kapitel 1 zusammengetragenen Stoffsammlung. Unter Berücksichtigung der Möglichkeiten des zu verwendenden Werkzeugs, in diesem Falle die hybride grafische Expertensystemschale aus Kapitel 3, wird eine gemäß Kapitel 4.2 zielgerichtete Wissensstrukturierung und Zuordnung zu den angebotenen Wissensrepräsentationformalisten durchgeführt.

Um einen möglichst allgemeingültigen Ansatz vorstellen zu können, soll eine Schale angenommen werden, die Mittel der Objektorientierung, der Regelmengenverarbeitung und der Überwachung von Zwängen besitzt. Sie erlaubt ereignisbasierte grafische Sitzungsabläufe und weist eine mächtige Erklärungs- und Hilfefunktion auf.

Im Rahmen der abschließend vorgestellten beispielhaften auf der Inferenzmaschine Babylon basierenden Implementierung wird schließlich eine Betrachtung durchgeführt werden, inwieweit diese Anforderungen an eine Schale in der Praxis eingehalten werden können.

#### **4.3.1 Objektorientierte Repräsentation des Faktenwissens**

Bei dem hier vorliegenden Auslegungsproblematik steht an einseitigem Faktenwissen in erster Linie die Materialparameter der verwendbaren Kerne zur Verfügung. Diese sind den Datenbüchern der Hersteller zu entnehmen. Aufgrund der großen Menge an Kernformen und ihren weitestgehend gleichen Merkmalen erscheint es sinnvoll, die Kerne als Objekte einzupflegen. Dazu wird ein Frame mit den kernbeschreibenden Merkmalen angelegt. Die einzelnen Kerne werden als Instanzen dieses Frames eingebracht. Als Merkmale können dann nicht nur die Herstellerdaten verwendet werden, sondern auch der jeweils anzuwendende Auslegungsalgorithmus.

Andererseits können gerade diese Materialdaten in einer Datenbank-anwendung gepflegt sein. Hier ist eine Schnittstelle zu einer solchen Anwendung denkbar. Durch geeignete Reporte können die Datenbankbestände exzerpiert werden und als ASCII-Dateien tabellarisch abgelegt werden. Diese können dann zum Sitzungsstart eingelesen werden und durch dynamische Instantiierung zur Laufzeit dem Expertensystem verfügbar gemacht werden.

Ein weiterer Vorteil ist, daß die Wissensbasis strukturiert auf mehrere Dateien verteilt wird. Dies erleichtert die weitere Pflege.

#### **4.3.2 Constraintbasierte Repräsentation des Kontrollwissens**

Die Constraints sind daher besonders zur Überwachung von Grenzfällen geeignet, da sie immer aktiv sind und im Hintergrund nach jeder Wertänderung abgearbeitet werden. Bezogen auf den Experten im vorliegenden Problem könnte damit zum Beispiel die Überwachung der Maximalinduktion assoziiert werden, die auch beim Experten nur dann wirklich aktiv wird, wenn er bei einer Rechnung eine Induktivität erhält, die über der zulässigen Maximalinduktion des berechneten Materials liegt.

Im Rahmen der allgemeinen Systembestimmung sollte der nicht erfahrene Benutzer des Expertensystems seine Anforderungen durch qualitative Aussagen spezifizieren können, insoweit es sich nicht um eindeutige Angaben wie die Ausgangsspannung handelt. Die Rahmenbedingungen der Netzteilauslegung sollen qualitativ berücksichtigt werden. Die Beachtung eines geringen Streufeldes kann je nach Anwendungsfall „absoluten Vorrang“ haben, sie kann aber auch „weniger wichtig“ sein. Derlei Eingaben des Benutzers sind zu überwachen, um sinnlose Aussagen als Anforderungsprofil zu vermeiden. Es kann nicht sein, daß bei gegenläufigen Abhängigkeiten unterschiedlicher Kenngrößen allen ein absoluter Vorrang zugeordnet wird. Auch dies ist ein Einsatzgebiet für Constraintnetze. Eine Plausibilitätsverletzung linguistischer Variablen entspricht einer Constraintverletzung. Tritt sie ein, kann ein Dialog mit dem Benutzer initiiert werden, der in einer Rücknahme beziehungsweise Änderung der verletzenden Eingabe mündet.

Constraintnetze erlauben die Propagierung eines nicht initialisierten letzten Werte des Netzes. Danach sind Wertänderungen immer so durchzuführen, daß keine Constraintverletzung eintritt. Daher erweisen sie sich in der Praxis als weniger geeignet, abhängige Größen im Zusammenhang mit der Änderung eines Wertes direkt mitzupflegen.

In objektorientierten Systemen ist der Zugriff auf die Merkmale eines Objekts nur indirekt über Methoden des Objekts möglich. Dies bedeutet, daß es Standardmethoden zum Lesen und Schreiben für jedes Merkmal des Objekts gibt. An den Enden der Constraintnetze hängen Merkmale von Objekten. Daher können Mechanismen, die mit Constraintnetzen realisiert worden sind, prinzipiell auch mit Methoden realisiert werden. Dies gilt es insbesondere zu prüfen, wenn die Fähigkeiten der Schale, bei Constraint-

verletzung zu reagieren, unzureichend sein sollten. Beispielsweise kann dies eine mangelhafte grafische Unterstützung sein.

### 4.3.3 Auslegungsalgorithmen und Bewertungsfunktion

Sofern eine exakte Formel für die Verluste im magnetischen Kern für ein bestimmtes Kernmaterial vorliegt und der thermische Widerstand des Transformator-kerns bekannt ist, können die Kernverluste exakt berechnet werden. Speziell für die Metallbandkerne der Firma VAC Vakuumschmelze und die EFD-Kerne der Firma Philips wurde die Ableitung dieser Berechnungsvorschrift im Kapitel 1.7.6.1 nachvollzogen.

Gesondert muß dabei lediglich die Maximalinduktion des Materials berücksichtigt werden. Sobald die mit den Gleichungen für die optimale Induktion berechnete Induktion über dieser Maximalinduktion liegt, ist die Maximalinduktion zu verwenden. Dann kann die Stromdichte nicht mehr nach der Gleichung für die optimale Stromdichte berechnet werden, da diese Gleichung nur gilt, wenn die Optimalinduktion verwendet wird. Die Stromdichte ergibt sich gemäß Gleichung 1.47.

Sobald die Induktion über der Maximalinduktion liegt, wird auch von der Regelmenge die Maximalinduktion verwendet und zur Berechnung der Stromdichte entsprechend die Gleichung 1.47. Die so berechnete Stromdichte wird dann auch in die Gleichung für die mit einem bestimmten Kern übertragbare Maximalleistung für  $J_{opt}$  eingesetzt. Diese Formel gilt allgemein. Die unterschiedlichen Schaltungsprinzipien werden in dem Faktor  $K$  berücksichtigt. Dieser Faktor wird für jedes Schaltungsprinzip gemäß [M2] berechnet.

Für die übrigen Ferritkerne existieren jedoch nur Nomogramme als Auslegungshilfe [M10]. Diese liegen in gedruckter Form vor (siehe Bild 1.17). Für eine rechentechnische Verarbeitung sind die für diskrete Scharparameter gezeichneten Kennlinien der abgebildeten Kurvenscharen in eine mathematisch geschlossene Form zu bringen. Dies geschieht mit Hilfe der linearen Regressionsanalyse in einem doppelt logarithmischen Koordinatensystem. Dies ist zulässig, da die Kennlinien im dargestellten Bereich immer annähernd linear sind. Für die Kennlinien jedes veröffentlichten Nomogramms muß diese Analyse einmal erfolgen, um ein mathematisch geschlossenes Abbild der Kurvenscharen bereitstellen zu können. Ansonsten muß der 6-schrittige Algorithmus für jeden Kern wie in Kapitel 1.7.6.2 beschreiben,



durchlaufen werden. In Schritt 4 entscheidet sich, ob ein auf seine Verwendbarkeit hin untersuchter Kern geeignet ist.

Für die übrigen Ferritkerne existieren jedoch nur Nomogramme als Auslegungshilfe. Diese liegen in gedruckter Form vor (Bild 1.17). Durch Regressionsanalyse sind die abgebildeten Kurvenscharen in mathematisch geschlossene Formen zu überführen. Für jedes veröffentlichte Nomogramm muß dies einmal erfolgen, um ein mathematisch geschlossenes Abbild der Kurvenscharen bereitstellen zu können. Ansonsten muß der 6-schrittige Algorithmus für jeden Kern wie in Kapitel 1.7.6.2 beschreiben, durchlaufen werden. In Schritt 4 entscheidet sich, ob der Prüfling besteht.

Bei Eintaktdurchfluß- und Sperrwandlern ist die sich so ergebende Lösungsmenge oft leer, denn die Beschränkung der Maximalinduktion verhindert vor allem bei kleinen Kernen den Einsatz bei niedriger Frequenz unter optimaler Induktion, und nur solche optimierten Auslegungen kann das Nomogramm ermitteln. Daher behilft man sich bei der Auslegung von Kernen für diese Wandler mit der Näherung, daß ein Kern, der bei der gleichen Induktion eingesetzt wird, bei der Hälfte der Schaltfrequenz etwa ein Viertel der Leistung übertragen kann. Es wird also im Anschluß an die Suche nach einem Transformator Kern, der bei optimaler Induktion betrieben werden kann, nach einem Kern gesucht, der bei doppelter Frequenz die vierfache Leistung übertragen kann, und dazu die entsprechende Induktion ermittelt. Die Induktion wird für jede so gefundene Lösung beibehalten, die alte Leistung wird eingesetzt und die gefundene Lösungsfrequenz für den speziellen Kern halbiert. Diese Werte werden dann in die Kern-Lösungsliste eingetragen.

Dieses Verfahren läßt sich entsprechend für weitere Vielfache fortsetzen. Das Vielfache der Primärleistung, nach der dann die Auslegung des Transformator Kerns vorgenommen wird, ist die Berechnungsleistung. Über eine Maximalfrequenz von 500 kHz ist jedoch keine gesicherte Auslegung mit den in [M10] enthaltenen Diagrammen möglich. Die maximale Auslegungsfrequenz muß daher auf 500 kHz beschränkt werden.

Die Auslegungen für Sperr- und Eintakt-Durchflußwandler unterscheiden sich nur geringfügig im Leistungsmaßstab der Übertragungsleistung.

Prinzipiell kann die im Kapitel 4.1 angedachte Bewertungsfunktion auf alle Ergebnisse des Kernauslegungslaufs angewandt werden, insbesondere wird sie leicht Kern-Topologie-Paarungen für ungeeignet erklären können,

die bereits früh an Ausschlußkriterien gescheitert sind. Andererseits bedeutet dies ein unnötiges Mehr an Rechenzeit, da der Benutzer in der Regel nur an den besten Lösungen interessiert ist. Eine heuristische Vorgehensweise empfiehlt daher, daß die Startmenge bereits frühzeitig verringert wird. Dann kann die abschließende Bewertung aber keine Aussagen mehr über den Ausschlußgrund mehr machen, weil ihr keine Daten darüber vorliegen.

Es sollte also zweischrittig implementiert werden:

- 1) In der vorgeschalteten Regelmenge zur Systembestimmung wird die Startmenge aufgrund der qualitativen wie quantitativen Vorgaben klein gewählt.
- 2) Nach Abschluß der Kernausslegung werden mittels der Bewertungsfunktion die Ergebnisse in einer Bestenliste einsortiert.

#### **4.3.4 Regelbasierte Repräsentation des Ableitungswissens**

Bei der Topologiefestlegung sollen neben quantitativen Aussagen auch qualitative Aussagen möglich sein. Die qualitativen Aussagen werden mittels linguistischer Variablen repräsentiert. Beispielsweise sind qualitative Aussagen zur Eignung von Kernbauformen in Bezug auf eine elektromagnetische Beeinflussung der Umgebung denkbar, da bei der Netzteilauslegung eine exakte Streufeldbetrachtung eher nicht die Regel sein wird. Derlei Betrachtungen können dazu genutzt werden, daß man unter kritischen Rahmenbedingungen von bestimmten Kernformtypen Abstand nimmt. Diese Prüfungen werden in das vorgeschaltete Regelwerk integriert.

In Kapitel 1 wurden gängige Verfahren zur Auslegung des Mittelfrequenztransformators für Schaltnetzteile vorgestellt. Diese unterscheiden sich in ihrer Einsetzbarkeit. Das in Kapitel 1.7.6.1 hergeleitete Verfahren läßt sich aufgrund des Bedarfs spezieller Formparameter nur bei Kernen einsetzen, für die diese Parameter vorliegen. Das Verfahren gemäß Kapitel 1.7.6.2 ist allgemeiner verwendbar, dafür aber nicht so exakt. In Anbetracht der hier betrachteten Kernformen und Materialien kann bereits folgendes grundsätzliches Vorgehen festgelegt werden: Kerne, für die die speziellen Parameter verfügbar sind, sollten mit den spezialisierten Auslegungsalgorithmen berechnet werden, alle anderen mit dem Näherungsansatz. Somit existiert derzeit kein Kern, der nicht auslegbar wäre.

Die Algorithmen zur Auswahl von Kern und Wicklung können prinzipiell von Regeln, Methoden und Constraints angestoßen werden. Dies ist im Rahmen der Constraintpropagierung eines nicht belegten Wertes wie die Windungszahl zwar prinzipiell möglich, aber aufgrund des schwerfälligen

Netzes unhandlich. Interessanter ist schon die Bindung eines Algorithmus als Behavior direkt an eine Instanz des Objekts Kern. Da aber auch die Topologie und andere Randbedingungen wie die thermische Wechselwirkung mit der Umgebung eine Rolle spielen, das Problem also eine höhere Zahl von zu prüfenden Abhängigkeiten aufweist, empfiehlt sich hier die Aktivierung aus einer Regel heraus.

Bezogen auf die in Kapitel 4.1 genannten Fragestellungen wurde das Wissen auf zwei Regelmengen verteilt.

Die einzelnen Vorgaben für eine Problemstellung werden allgemein als Teilaspekte bezeichnet. Die Werte der Teilaspekte können nach dem Durchlaufen der Regelmenge verändert werden, um den Problemlösungsprozeß erneut zu starten. Dadurch können leicht mehrere Varianten eines Problems berechnet werden. Das Wissen über die für die Lösung des Problems notwendigen Teilaspekte ist in diesen Instanzen gebunden.

#### **4.3.4.1 Regelmenge zur Systembestimmung**

Die Regelmenge zur Systembestimmung hat primär die Aufgabe, das Schaltungsprinzip des Schaltnetzteils zu bestimmen. Das Regelwerk wird aufgerufen, wenn der Benutzer die Problemstellung im Sitzungsverlauf wählt. Innerhalb der Regelmenge wird nun, in Abhängigkeit von den Voreinstellungen der Teilaspekte durch den Benutzer, eine Auswahl zwischen verschiedenen Lösungsmöglichkeiten getroffen.

Abhängig von der eingegebenen Sekundärleistung, der Höhe der Primärspannung und der eingestellten Art der Schaltventile wird die obere Grenzfrequenz beziehungsweise die Art der einsetzbaren Schaltventile für diese Problemstellung gewählt. Ist eine bestimmte Ventilart vorgegeben, die für diese Problemstellung nicht einsetzbar ist, wird sie durch eine andere Art von Schaltventilen ersetzt [M15, M16, M30]. Der Benutzer wird über diese Veränderungen durch Meldungen informiert.

Die Wichtung des Streufeldes wird für eine Einschränkung der Kernformauswahl genutzt. Je nach Wichtung wird das Spektrum der möglichen Kernformate entsprechend den in [M20] genannten Untersuchungsergebnissen eingeschränkt.

Gibt der Benutzer ein Kernmaterial fest vor, wird die Schaltfrequenz für dieses Problem je nach Material entsprechend begrenzt oder beibehalten.

So wird das Problem in einzelne Lösungsschritte unterteilt, die nacheinander abgearbeitet werden können. Treten Inkonsistenzen auf, wie zum Beispiel bei der Vorgabe einer hohen Spannungsdifferenz zwischen Primär-

und Sekundärwicklung, einer Isolationsklasse für den Transformator bei einer gleichzeitigen hohen Wichtung für das Streufeld des Transformators, wird dies dem Benutzer angezeigt. Er hat dann die Möglichkeit, einer Rücknahme einzelner Vorgaben zuzustimmen. Stimmt der dem zu, wird die Vorgabe geändert. Die Regelmenge führt dann alle Regeln noch einmal aus, die diese Vorgabe berühren. Lehnt der Benutzer eine Änderung ab, wird die Abarbeitung der Regelmenge abgebrochen. Es erfolgt eine entsprechende Ausgabe.

Um dieses Verhalten zu realisieren, ist eine wiederholte sequentielle Abarbeitung unter Festlegung eines Abbruchkriteriums ist empfohlen. Dadurch wird erreicht, daß das Prüfkriterium im Anschluß an jede ausgeführte Regel überprüft wird. So ist jede Regel in der Lage, die Abarbeitung der weiteren Regeln bei Inkonsistenzen zu verhindern.

Die Regelmenge wird mit der Ausgabe der vorzugebenden Grenzfrequenz, Ventilart, Schaltungsprinzip, Leistungsklasse Kernformbegrenzung, Kernmaterialbegrenzung und Isolationsklasse abgeschlossen. Diese Daten können als Voreinstellungswerte für die Regelmenge Kernausslegung verwendet werden.

#### **4.3.4.2 Regelmenge zur Kernbestimmung**

Die Regelmenge Kernausslegung wird entweder angestoßen, nachdem alle Teilaspekte befüllt worden sind. Dies kann durch den Benutzer direkt, oder als Ergebnis der Systembestimmung erfolgen.

Der Ablauf der Lösungsfindung gleicht in der Regelmenge Kernbestimmung dem Ablauf bei der Regelmenge Systembestimmung. Hier kann die Aktivierung ganzer Regelblöcke durch eine bestimmte Regel mit wahrer Regelprämisse über eine Flaggen-Steuerung stärker genutzt werden. Dies ist nötig, um eine Unterscheidung zwischen den beiden Möglichkeiten des Anstoßens dieser Regelmenge zu treffen. Wird die Regelmenge aus der Problemstellung Kernausslegung heraus angestoßen, ist zum Beispiel die Vorgabe „sonstige Materialien“ für den Teilaspekt Materialvorgaben anders zu verarbeiten als dieser Regelmenge über die Problemstellung Systembestimmung.

Viele Prüfungen, die die Regelmenge System durchführt, können hier entfallen, um Rechenzeit zu sparen. Es wird beispielsweise nicht mehr geprüft, ob für die gewählte Grenzfrequenz überhaupt ein Schaltventil existiert, das zur Übertragung der eingestellten Leistung bei der eingestellten Zwischenkreisspannung und dem gegebenen Schaltungsprinzip geeignet ist.

Die Regeln enthalten jeweils nur Teile des Wissens aus der Stoffsammlung. Bei der Berechnung der Kerne wird zunächst zwischen VAC-Metallbandkernen, EFD-Kernen mit den Materialien von Philips und übrigen Kernformen und Materialien unterschieden. Die Unterscheidung zwischen diesen Kerntypen ist nötig, da für die VAC-Metallbandkerne und die EFD-Kerne mit Philipsmaterialien exakte Formeln für die Berechnung der Kernverluste und des thermischen Widerstandes der Kerne vorliegen, für die übrigen Ferritkerne jedoch nur Nomogramme existieren.

Die Regelmenge ist so aufgebaut, daß die Regeln für die exakte Berechnung der Metallbandkerne der Regeln für Sperr, Eintakt- und Brückenwandler vorgeschaltet sind. Bei der Vorgabe Sperrwandler als Schalungsprinzip reagiert diese Regel für VAC-Kerne nicht, da es nicht möglich ist, in einen Ringkern den für einen Sperrwandler bei hochpermeablem Material zur Speicherung der Energie notwendigen Luftspalt einzufügen. Die Menge der zur Auswahl stehenden Kernformen wird dann, wenn eine Regel zur exakten Kernberechnung reagiert hat, entsprechend eingeschränkt. Dadurch werden Doppelauslegungen vermieden. Nachfolgend wird immer genau eine der Nomogramm-Regeln reagieren und eine vollständige Kernausswahl bereitstellen. Am Ende dieser Berechnungen wird überprüft, ob für die jeweilige Kernform in den dynamischen Instanzen ein Kernmaterial für den betreffenden Kern eingetragen ist, das bei der ermittelten Übertragungsfrequenz eingesetzt werden kann. Wird kein Material gefunden, erhält der Benutzer eine Mitteilung darüber. Werden ein oder mehrere Materialien gefunden, die für die jeweilige Übertragungsfrequenz geeignet sind, werden sie in eine Liste eingetragen, die der jeweiligen Kernkennung angefügt und wieder in einer Kernlösungsliste gespeichert wird.

Diesem Berechnungszyklus folgt die Möglichkeit, die Anzahl der Kerne, die weiterbetrachtet werden sollen, selbst entsprechend einzuschränken. Neben der Kernform wird angezeigt, bei welchem Induktionshub und bei welcher Übertragungsfrequenz der jeweilige Kern eingesetzt werden soll. Die Kerne sind in eine Liste eingetragen, aus der der Benutzer auswählen kann. Die restlichen Kerne werden nicht weiter betrachtet.

Danach folgt eine Überprüfung der verbliebenen Kerne auf bestimmte Kernmaterialvorgaben. Spezielle Vorgaben für das Kernmaterial führen zu einer erheblichen Einschränkung der Kernausswahl. Daher sollten zunächst keine Materialvorgaben gemacht werden. Die möglichen Kernmaterialien werden bei der Lösungsausgabe in jedem Fall angezeigt. Der Benutzer kann dann noch wählen.

Daran anschließend folgt eine Gruppe von Regeln, die die Kernlösungsliste nach den eingestellten Gewichtungen für Masse, Größe und Streufeld sortieren. Sofern einer dieser Teilaspekte mit dem Wert „absoluter Vorrang“ belegt ist, wird die Lösungsliste nur nach dieser Vorgabe sortiert. Ist ein Unterschied zwischen den Wichtungen ein oder zwei Stufen darunter festzustellen, wird auch das in der Sortierung berücksichtigt. Es wird dann zunächst nach einer der Vorgaben Masse oder Größe sortiert und dann nach der Liste der Kernformenreihenfolge für ein minimales Streufeld oder umgekehrt. Dadurch wird garantiert, daß immer eine sortierte Liste vor dem Beginn der Ausgabe vorhanden ist, die den Auswahlkriterien des Benutzers entspricht. Der erste Kern der Lösungsliste ist immer derjenige, der die vom Benutzer eingestellten Bedingungen am besten erfüllt.

Bei dem bis hierhin vorgestellten Lösungsweg kann es zu Abweichungen gegenüber dem optimalen Übersetzungsverhältnis bei der Berechnung der Windungszahlen für Primär- und Sekundärwicklungen kommen. Daraus würde eine Veränderung des eingestellten stationären Tastverhältnisses resultieren und damit eine Absenkung des Induktionshubs. Die berechnete Übertragungsleistung wäre damit aber nicht mehr übertragbar. Die Abweichungen vom Übersetzungsverhältnis sind zum Teil darauf zurückzuführen, daß bei Rundungen zu starke Fehler auftreten. Rundungen sind hier aber unumgänglich, da nur ganze Windungszahlen für die Kopplung mit einem magnetischen Kreis möglich sind. Zum anderen ist eine feste Vorgabe für die Primärwindungszahl notwendig, damit bei minimaler Zwischenkreisspannung noch das maximale Tastverhältnis des Schaltungsprinzips ausgenutzt werden kann, ohne den Kern in die Sättigung zu fahren und damit die Schaltventile durch Überstrom zu zerstören. Das maximale Tastverhältnis ist in diesem Moment aber sehr wichtig, um die Sekundärspannung überhaupt noch garantieren zu können. Daher ist es notwendig, diese Rundungen und den Fehler zwischen tatsächlichem Übersetzungsverhältnis und dem über Zwischenkreisspannung, stationäres Tastverhältnis und Sekundärspannung vorgegebenen Übersetzungsverhältnis zu überwachen. Dazu dient eine Regel, die die beiden möglichen Rundungen der primären Windungszahl mit den vier möglichen Rundungen der sekundären Windungszahl kombiniert. Von diesen Kombinationen wird diejenige gewählt, die gegenüber dem vorgegebenen Übersetzungsverhältnis die geringste Abweichung aufweist. Ferner wird kontrolliert, ob der sich so ergebende Fehler unter 9% liegt, da dies der Toleranz entspricht, die üblicherweise bei der Auswahl des stationären Tastverhältnisses eingeräumt wird. Erfüllt der Fehler diese Vorgabe nicht,

wird der Kern aus der Liste gestrichen. Dem Benutzer wird darüber eine Mitteilung gemacht. Wenn der Toleranzbereich eingehalten wird, wird das Übersetzungsverhältnis an die Kernform gebunden und in die Kernlösungsliste übernommen.

Bei Sperrwandlern wird die Windungszahl etwas anders berechnet als bei den übrigen Schaltungsprinzipien. Diese Unterscheidung ist hier genauso berücksichtigt wie die Berechnung des nötigen Luftspalts. Diese Daten werden ebenfalls in der Liste festgehalten. Damit ist die Liste für die Ausgabe fertiggestellt.

Die Ausgabe wird durch eine Regel gesteuert, die je nach Schaltungsprinzip und Anzahl der Sekundärkreise die Formatierung der Ausgabe ändert. Ferner überprüft diese Regel den Wickelraum des Kerns. Ist der nicht ausreichend, wird der Kern aussortiert. Hierüber wird ebenfalls eine Mitteilung auf dem Bildschirm erscheinen. Berechnet wird auch der maximale Drahtquerschnitt, der aufgrund des Skineffekts frequenzabhängig ist. Die Anzahl der nötigen parallelen Wicklungen wird ebenfalls ausgegeben.

Parallelwicklungen sind bei der Überprüfung des Wickelraums ebenfalls zu berücksichtigen.

Für Eintaktdurchflußwandler wird die Windungszahl der Entmagnetisierungswicklung über das eingestellte stationäre Tastverhältnis berechnet.

Für die Kerne die diese Prozedur überlebt haben, können nun jeweils die Berechnungsergebnisse ausgegeben werden. Weiterhin läßt sich die Bewertungsfunktion auf diese Ergebnisliste anwenden. So hat der Benutzer die Möglichkeit, verschiedenen Kerne zu vergleichen.

#### **4.3.5 Benutzerschnittstelle mit grafischer Erklärungskomponente**

Wie bei anderen grafischen Anwendungen werden Eingaben des Benutzers dem System über Eingabefelder, Druckknöpfe oder Auswahldialoge realisiert. Ausgaben finden auf dem Bildschirm durch Darstellung von Bildern oder als ein- oder mehrzeiliger Text statt.

Ein Expertensystem bietet den zusätzlichen Komfort, jederzeit über den Fortschritt des Inferenzprozesses abgefragt zu können. Dafür ist es eine Erklärungskomponente vorgesehen. Die Leistungsfähigkeit der Erklärungskomponente definiert sich folgendermaßen: Zu jedem Zeitpunkt der Problemlösung läßt sich die Erklärungskomponente anstoßen. Beispielsweise nach Ende der Abarbeitung der Regelmenge in Form eines Druckknopfs auf der Oberfläche, die die gesamte Entscheidungshistorie aufzeichnet **oder** nur

die der ausgeführten Regeln **oder** immer dann, wenn eine Entscheidung ansteht.

Leider bieten diese Erklärungskomponenten in der Regel nur Aussagen, die dem Wissensingenieur bei der Implementierung helfen. Diese Informationen sind von der Art:

*Die Regel ~a wird [nicht] ausgeführt, weil die maximale Grenzfrequenz der gewählten Schaltventile (~a Hertz) [kleiner / größer gleich] ist als die geforderten ~a Hertz [und ...]*

Die Platzhalter werden dabei mit aktuellen Werten befüllt.

Der Benutzer hingegen erhofft sich eine auf seinen Erfahrungshorizont sowie die Aufgabenstellung des Expertensystems bezogenen optimierte Metahilfe, die die Abläufe im System mit dem nötigen Hintergrundwissen füllen. Das können Entscheidungshilfen **vor** der Eingabe von Werten sein. Bei Rückfragen wird über die Auswirkungen informiert und warum man bis hierhin gekommen ist. Dazu werden gegebenenfalls Texte oder Grafiken zur Erläuterung hinzugezogen.

Hier würde also eine Klartexthilfe im Dialogfenster oder als Knopf auf dem Dialogfenster reichen, der die anstehende Entscheidung erläutert. Diese kann gegebenenfalls umfangreicher ausfallen, indem der Hilfetext mit Aktualparametern angereichert wird oder indem wieder ein Verweis auf die maschinelle Entscheidungshistorie (beispielsweise als Knopf „erweiterte Information“) angeboten wird.

Eine den Bedürfnissen des Benutzers gerechtwerdende Hilfefunktionalität sollte aufgrund der Datenverwaltung in der hybriden Schale an die Objekte gebunden werden und mit den Mitteln der grafischen Oberflächengestaltung realisiert werden.

#### **4.4 Implementierung eines Expertensystem zur Auslegung des Mittelfrequenztransformators primärgetakteter Schaltnetzteile**

Am Lehrstuhl für elektrische Maschinen und Antriebe wurde ausgehend von den in Kapitel 4 erarbeiteten Maßgaben zur Wissensstrukturierung unter dem Namen TRAFEX eine Testimplementation vorgenommen. Das Expertensystem basiert auf der am Lehrstuhl entwickelten und in Kapitel 3 kurz vorgestellten hybriden grafischen Expertensystemschale. Bei der Implementierung fanden die Möglichkeiten dieser Schale Berücksichtigung. Insofern konnte es zu geringfügigen Abweichungen von den hier vorgestellten



ten Wissensstrukturierung kommen. Diese sind im wesentlichen auf die Realisierung einer grafischen Benutzerschnittstelle und die vorgefundenen Gestaltungsmöglichkeiten beschränkt. Die Auslegungsphase mehrerer für den Laborbetrieb gebauter Schaltnetzteile ist von Sitzungen mit diesem Expertensystem begleitet worden.

#### 4.4.1 Wissensstrukturierung

Eine ausführliche Diskussion der Eignung unterschiedlicher Wissensrepräsentationsformen hybrider Schalen ist wiederholt durchgeführt worden [X1-X7]. Eine zusammenfassende Bewertung der jeweiligen optimalen Einsatzgebiete ergibt sich im Hinblick auf die Aufgabenstellung sowie Stoffsammlung gemäß Kapitel 1 wie folgt:

- Die Datenhaltung (Faktenwissen) ist abbildbar auf Instanzen von Frames. Eine Vererbung ist möglich, aber nicht ergiebig.
- Abhängigkeiten zwischen Größen sind gut über Constraintnetze, aber auch über Behaviors realisierbar.
- Die Auslegungsalgorithmen können aus Behaviors, Constraintsnetzen oder Regeln heraus aktiviert werden

Die Einsetzbarkeit von Constraintnetzen ist unter Babylon eingeschränkt, weil die Netze nach jeder Propagierung immer wieder zurückgesetzt werden müssen, die abhängigen Variablen also explizit wieder in einen uninitialisierten Zustand zu überführen sind. Eine Kontrolle entarteter Variablen über Schreib- und Lesebehaviors erscheint besser geeignet

Das Fachwissen von TRAFEX ist unter Zuhilfenahme von drei Wissensrepräsentationsformalismen eingebracht worden.

- **Fakten** beinhalten das Wissen über die Eigenschaften von Objekten. Bei TRAFEX werden die spezifischen Eigenschaften der Kerne, wie ihre Geometriedaten und die Materialien, in denen eine Kernform verfügbar ist, als Faktenwissen repräsentiert. Derartige Informationen können auch in Datenbanken abgelegt werden. TRAFEX liest diese Fakten aus Textdateien, wenn die Sitzung gestartet wird, und speichert sie in dynamisch erzeugten Instanzen. Auf diese Weise können neue Produkte leicht hinzugenommen werden, ohne die Wissensbasis selbst ändern zu müssen. Derzeit sind über 130 Kerngeometrien für 13 Kerntypen wie E-, EFD- und P-Kerne, sowie Ring- und Schnittbandkerne, und 10 Kernmaterialien von Philips und Siemens wie N27, 3C80 und 3F3 verfügbar [T6, T10]. Zusätzlich kann auf die Produkte von VAC zurückgegriffen werden [T9].

- **Beziehungen** beinhalten das Wissen um Abhängigkeiten zwischen einzelnen Eigenschaften unterschiedlicher Komponenten und die sich ergebenden Auswirkungen auf diese Größen bei Veränderungen einer einzelnen von ihnen. Sie werden in Behaviors abgebildet. Der objektorientierte Wissensrepräsentationsmechanismus stellt diese zur Verfügung.
- Die Aktionen beziehungsweise **Vorgehensweisen** des Fachmanns, die schlußendlich zur Problemlösung führen, werden in Regeln formuliert und schließlich in Regelmengen themenbezogen zusammengefaßt.

Auf Constraintnetze ist wegen der mangelhaften grafischen Unterstützung bei der Aufhebung von Constraintverletzungen verzichtet worden. Stattdessen sind ähnlich geeignete Methoden realisiert worden.

Eine detaillierte Beschreibung der vorgenommenen Implementierung der Wissensbasis TRAFEX für die Schale EMA-XPS steht in [T13] zur Verfügung.

#### 4.4.2 Realisierung einer Oberfläche mit grafischem Hilfesystem

Mit den Mitteln der Expertensystemschaale EMA-XPS kann die notwendige Funktionalität zur Kontrolle der Benutzerschnittstelle in die Wissensbasis eingebracht werden. Es sind Antworten auf die möglichen Ereignisse, wie das Betätigen der Druckknöpfe und die Auswertung der Rückgaben der Dialogfenster, formuliert worden. Zur Optimierung der Ergonomie und um eine Überfrachtung des Sitzungsschirms zu verhindern, werden nur wenige grafische Elemente kreiert. Dafür aber werden diese im Verlauf einer Sitzung unterschiedlich genutzt.

Bei den meisten erhältlichen Expertensystemschaalen ist die Leistungsfähigkeit der vorhandenen Erklärungskomponente unbefriedigend [T4]. Daher ist zusätzliche Funktionalität zur Bereitstellung von Erklärungen und Hilfestellungen in Form von Texten und Grafiken realisiert worden. Diese Mittel informieren den Benutzer über den aktuellen Zustand des Inferenzprozesses, sofern es erforderlich ist.

Der Bildschirm ist in einen linken Eingabeteil und einen rechten Anzeigeteil unterteilt. Der Eingabeteil ist so aufgebaut, daß es später einfach ist, neue Arten von Teilaspekten hinzuzufügen, wenn eine Regelmenge erweitert werden soll. Die einzelnen Teilaspekte sind als Instanzen des Frames TEILASPEKT gebunden. Die Teilaspekte können verschiedenen Fragestellungen zugeordnet werden, da der Benutzer immer nur eine Fragestellung aktivieren kann. Sie werden immer auf der linken Bildschirmseite darge-

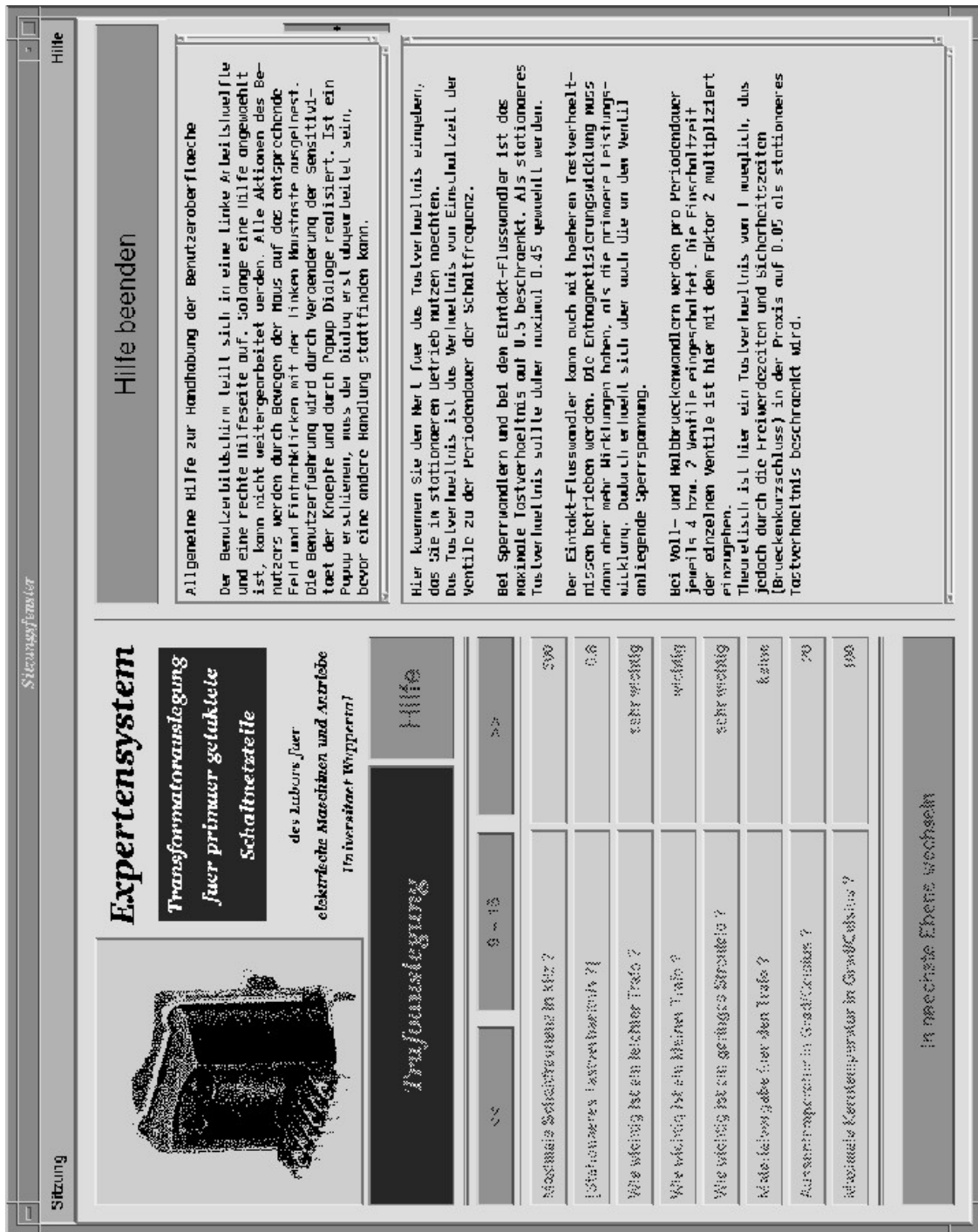


Bild 4.2: Der Endbenutzer-Bildschirm von TRAFEX während der Eingabephase des Anforderungsprofils für die Transformatorauslegung

stellt. Es werden nur die Teilaspekte angezeigt, die in dem gewählten Lösungsablauf verarbeitet werden. Diese Teilaspekte werden jeweils auf zwei Schaltflächen dargestellt. Die linke Schaltfläche ist mit der Benennung des Teilaspektes beschriftet, die rechte Schaltfläche mit dem aktuell einge-

stellten Wert dieses Teilaspektes. Eine Erläuterung zu den jeweiligen Teilaspekten erhält der Benutzer, wenn er die Schaltfläche mit der Benennung des Teilaspekts anwählt. Der Wert des Teilaspekts wird durch Anwählen der rechten Schaltfläche geändert. Ein Dialogfenster erscheint, in dem, bezogen auf diesen Teilaspekt, die möglichen Einstellungen dargestellt werden. Zum Bildschirmaufbau, den einzelnen Fragestellungen und dem Lösungsablauf der einzelnen Fragestellungen kann der Benutzer über eine Hilfeschnittfläche einen Hilfetext anfordern. Dieser Hilfetext ist in einer Textdatei gespeichert und wird vom Expertensystem nur dann geladen, wenn der Benutzer ihn anfordert. Angezeigt wird dieser Hilfetext auf der rechten Bildschirmseite. Die rechte Bildschirmseite wird weiterhin für die Ergebnisausgabe genutzt. Die Ergebnisausgabe beinhaltet auch die Anzeige von Grafiken. Diese Bilder stellen zum einen das Schaltungsprinzip, zum anderen den Kerntyp dar. Sie werden auf der rechten Bildschirmseite oben angezeigt. Unten erfolgt die Textausgabe mit den für den Benutzer wichtigen Angaben für Kern und Wicklungen des Leistungstransformators (Bild 4.2).

#### 4.4.3 Dokumentation eines Sitzungsablaufs

Zum Nachweis der Leistungsfähigkeit der in Kapitel 4.3 vorgestellten Konzepts zur Wissensimplementierung soll am Beispiel von TRAFEX der Ablauf einer kompletten Transformatorauslegung nachvollzogen werden. Zunächst wird ein geeignetes Schaltnetzteilprinzip ermittelt.

Effektivwert der Eingangsspannung:	230 V
Güte der Ausgangsspannung:	hoch
Anzahl der Sekundärkreise:	1
Gleichspannungswert am Ausgang:	12 V
Leistungsabgabe:	500 W
Geringes Streufeld:	sehr wichtig
Kleine Bauform:	wichtig
Leichter Transformator:	sehr wichtig
Kernmaterialvorgaben:	keine
Montageort:	frei auf der Leiterplatte
Isolationsklasse:	keine Vorgaben
Leistungshalbleiter:	keine Vorgaben

Tabelle 4.1: Anforderungsprofil an ein zu erstellendes Schaltnetzteil

Im Anschluß daran werden der beste Kern und die entsprechende Wicklungsanordnung bestimmt. Es wird angenommen, daß ein zu entwickelndes Schaltnetzteil die in Tabelle 4.1 aufgeführten Spezifikationen zu erfüllen habe.

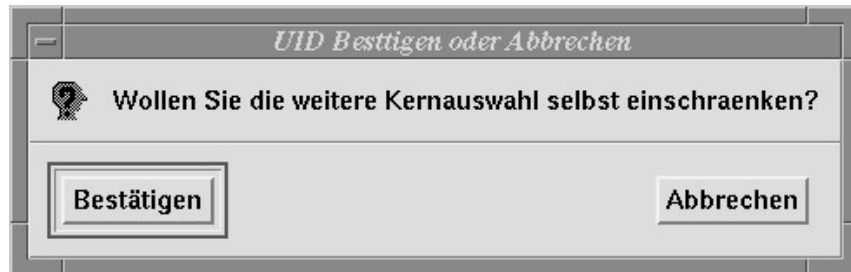


Bild 4.3: Dialogfenster, das vom Benutzer das weitere Vorgehen erfragt

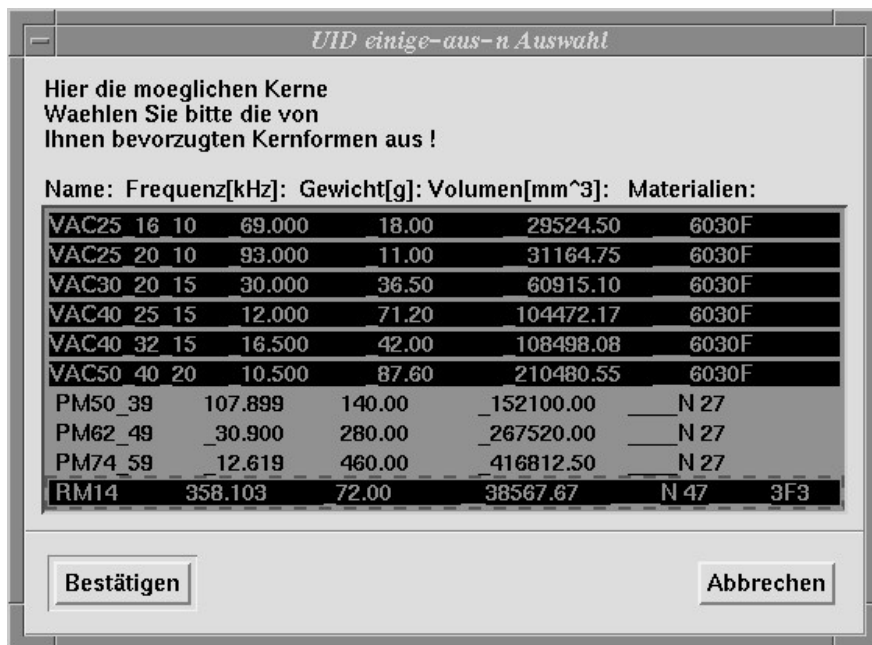


Bild 4.4: Mehrfachauswahldialog

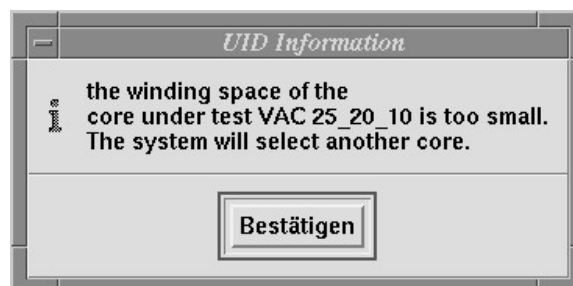


Bild 4.5: Benutzerinformation mit unterschiedlichen Spracheinstellungen bei EMA-XPS und TRAFEX

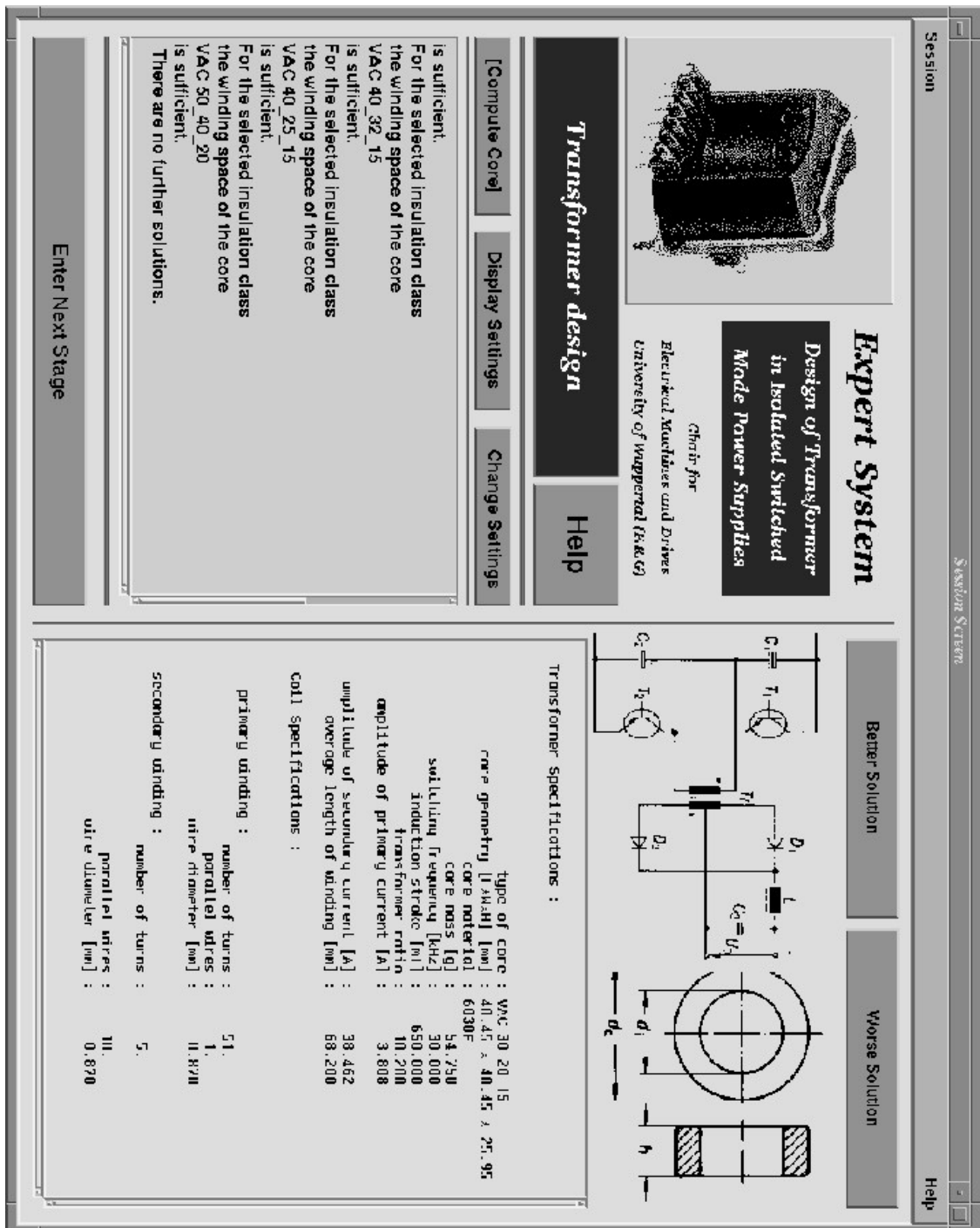


Bild 4.6: Präsentation der Ergebnisse der Transformatorauslegung in englischer Sprache

Diese Vorgaben werden zu Beginn der Topologieauswahl-Sitzung von TRAFEX eingegeben, um das geeignetste Schaltungsprinzip zu bestimmen. Erfahrene Benutzer können diesen Schritt überspringen und direkt mit der

Auslegungs-Sitzung beginnen. Wenn mit der Topologie-Stufe begonnen wird, startet diese die Abarbeitung der ersten Regelmenge und präsentiert schließlich die ermittelten Schlußfolgerungen gemäß Tabelle 4.2.

Im Anschluß an diese Stufe wird in die Auslegungssitzung gewechselt. Die voreingestellten Werte für die Auslegungsvorgaben werden mit den ermittelten Resultaten der Topologieauswahl-Sitzung überschrieben. Bild 4.4 zeigt den Hauptbildschirm von TRAFEX in dieser Stufe. Durch Betätigen der Knöpfe mit den Namen der für die Berechnung relevanten Merkmale können jeweils Informationen und Hilfestellungen abgefragt werden. Wertänderungen können durch Betätigen des zugehörigen Werte-Felds initiiert werden. Dazu erscheinen geeignete Popupdialoge.

maximale Schaltfrequenz:	500 kHz
Leistungshalbleiter:	MOSFET
Schaltungsprinzip:	Halbbrücken-Durchflußwandler
Leistungsbereich:	mittel
gewählte Kernformen:	(P PM RM R VAC)
ausgeschlossene Kernmaterialien:	keine
Isolationsklasse:	keine

Tabelle 4.2: Ergebnisse der Topologie-Stufe

Im Anschluß daran wird mit der eigentlichen Transformatorauslegung begonnen. Falls gewünscht, kann sich der Benutzer bereits im Vorfeld auf bestimmte Kerntypen festlegen. Andernfalls fällt das System diese Entscheidung. Immer dann, wenn die Inferenzmaschine mit dem Benutzer in Verbindung treten will, erzeugt sie Dialogfenster, wie die in den Bildern 4.3-4.5. Die Wärmeentwicklung und die Möglichkeiten der Wärmeabführung, hierbei auch der Montageort des Transformators, werden überprüft. Zum Ende der Auslegungssitzung werden die ermittelten Kernspezifikationen und die Wicklungsvorschriften präsentiert. Dabei werden unterschiedlich gute Auslegungen zur Wahl angeboten. Bild 4.6 zeigt den Sitzungshauptbildschirm in dieser Stufe, wobei EMA-XPS und TRAFEX für englische Ausgaben voreingestellt worden sind.

## 4.5 Zusammenfassung

Die Wissensbasis TRAFEX belegt, daß es möglich ist, auch sehr in die Tiefe gehende Problemstellungen, die rechenintensiv sind, in Expertensysteme einzubringen. Der Zeitaufwand für dieses Einbringen steigt allerdings enorm mit zunehmender Komplexität der Aufgabenstellung. Die eigentliche Stärke des Expertensystems bleibt auch hier die Möglichkeit, vage Entscheidungskriterien handhaben zu können und Entscheidungsprozesse eines Experten regelbasiert nachzubilden. Es ist damit gelungen, eine Wissensbasis zu erstellen, die dem Benutzer schnell den Umgang auch mit komplexen Kriterien bei der Auslegung von Transformatoren für Schaltnetzteile und bei der Auswahl von Schaltnetzteilprinzipien ermöglicht.

Um Akzeptanz für die betriebliche Einführung eines derartigen Softwarewerkzeugs zu schaffen, sind aber noch grundsätzliche Probleme zu klären. Eine Software lebt nur so lange, wie sie gepflegt wird. In diesem Fall bedeutet dies, daß mit dem Aufkommen neuer Kerne oder der Veröffentlichung präziserer Auslegungsalgorithmen dieses Werkzeug aktualisiert werden muß. Solange nicht erwartet werden kann, daß die Hersteller von Magnetmaterialien gleich Erweiterungsmodule mitanbieten, ist dieses neue Wissen vom Benutzer selbst hinzuzufügen. Der vorgestellte modulare Ansatz unterstützt hierbei in unterschiedlicher Weise. Erweiterungen der Kernparameterdateien sind sicherlich am leichtesten vorzunehmen. Das Einpflegen neuer Algorithmen bedeutet eine Programmierung in der zugrundeliegenden Wissensimplementationsprache. Allgemein findet die Notation des hier verwendeten LISP-Interpreters wenig Anklang. Die **Wissenserwerbskomponente** muß ausgebaut werden. Spezialisierte grafische Editoren sollten eine Programmierung in einer Metasprache ermöglichen. Die unterliegenden Mechanismen sollten für den Benutzer unsichtbar bleiben. Im Fall geschlossener mathematischer Gleichungen könnten grafische Formeleditoren zum Einsatz kommen. Kennlinienverläufe sind als Vektorgrafik in grafische Kennlinieneditoren oder Zeichenprogrammen interaktiv veränderbar. Die Ablauflogik sollte in einer einfachen Sprachnotation gehalten sein. Die gesamte Eingabe könnte demgemäß im Präsentationsstil zeitgemäßer DTP-Textverarbeitungen geschlossen stattfinden.



## 5. Schlußwort

Im Rahmen dieser Arbeit konnte gezeigt werden, daß die Auslegung von Transformatoren für Schaltnetzteile ein umfangreiches Unterfangen darstellt, bei dem viele Einflußgrößen in wechselseitiger und vielfach nichtstetiger Abhängigkeit zueinander stehen. Eine derartige Aufgabe ist geeignet, in Form eines Expertensystems behandelt zu werden.

Analog dem Vorgehen eines mit der Wissensimplementierung befaßten Softwareingenieurs wird ein Konzept für eine Teilautomatisierung der Aufgabe mit dem Ergebnis einer starken Entlastung des Schaltnetzteilentwicklers aufgezeigt. Dialoggesteuert wird zunächst mit Unterstützung aus der Anwendung das Anforderungsprofil an das zu realisierende Netzteil eingepflegt. Daraufhin wird die Netzteiltopologie bestimmt. Auf der Basis einer Kerndatenbank werden Kandidaten ermittelt, sowie Wickelvorschriften ausgegeben. Diese Ergebnisse lassen sich gegeneinander bewerten. Nötigenfalls lassen sich durch eine Anpassung der Startbedingungen befriedigendere Ergebnisse erzielen.

Basierend auf diesem Ansatz ist die Expertensystemschale TRAFEX entwickelt worden, um bei der Auslegung des Mittelfrequenztransformators für eine gegebene Schaltnetzteiltopologie zu unterstützen. TRAFEX wählt aus einer Datenbank den optimalen Kern für das zu entwickelnde Schaltnetzteil aus. Dabei werden Kriterien wie die Leistungsbemessung, Ausgangsspannung, thermische Bemessung, das Bauvolumen und die Isolationsklasse berücksichtigt. Zur Ermöglichung benutzerfreundlich gestalteter interaktiver Sitzungen ist eine grafische Benutzerschnittstelle erstellt worden. Anhand einer Beispielsitzung konnte die Flexibilität und die einfache Handhabbarkeit des Expertensystems TRAFEX unter Beweis stellen, das unter der hybriden grafischen Expertensystemschale EMA-XPS abläuffähig ist.

Wissensbasierte Systeme sowie Werkzeugumgebungen für deren Erstellung existieren seit Jahren in unterschiedlichen Ausprägungen. Im Rahmen dieser Arbeit wurde darüberhinaus ein Ansatz vorgestellt, der es ermöglicht, eine leistungsfähige Wissensverarbeitungseinheit mit geringfügigem Änderungsaufwand um eine grafische Mensch-Maschine-Schnittstelle zu erweitern. Die Geringfügigkeit des Aufwands bezieht sich auf die Änderungen an der wissensverarbeitenden Einheit.

Der Ansatz basiert auf einer nebenläufigen Aufgabenteilung zwischen Grafikanwendung und Wissensverarbeitung. Er setzt präemptive Multitas-

kingbetriebssysteme voraus. Dieser Ansatz ist verallgemeinerbar. Durch ihn läßt sich eine Vielzahl im Einsatz befindlicher komplexer Anwendungen ohne aufwendige Quelltextreorganisation grafikfähig machen.

Anhand der Schale EMA-XPS konnte die Leistungsfähigkeit eines nebenläufigen Softwarekonzepts unter Beweis gestellt werden, das eine aufwendige Restrukturierung großer Softwareanwendungen zum Zwecke der Erweiterung um eine grafische Benutzerschnittstelle überflüssig macht.

In Kapitel 1 wurden die Vorgehensweisen für die optimale Auslegung von Mittelfrequenztransformatoren vorgestellt. In Kapitel 2 wurde nachgewiesen, daß der Entwicklungsingenieur im Rahmen der Auslegung von einem Expertensystem, entwickelt unter einer hybriden grafischen Expertensystemschaale, gut unterstützt werden kann. In Kapitel 3 ist schließlich ein Konzept zur aufwandsarmen Erstellung einer leistungsfähigen hybriden grafischen Expertensystemschaale erläutert worden. In Kapitel 4 ist das Konzept für die Wissensimplementierung vorgestellt worden. Weiterhin ist die Implementierung mit dem Expertensystem TRAFEX auf der Basis der Expertensystemschaale EMA-XPS vorgestellt worden. Eine Beispielsitzung mit TRAFEX beschließt die Arbeit.

## 6. Schrifttum

Die im folgenden aufgeführte Literatur ist zunächst nach Themengebieten sortiert. Dies ist nach Möglichkeit auch innerhalb der Themengebiete aufrechtgehalten worden.

### **Mittelfrequenztransformatoren:**

- [M1] Siemens AG, „Ferrite und Zubehör Datenbuch 1990/91“, Siemens AG, 1989
- [M2] Vakuumschmelze Hanau, „Approximation formula for the maximal transformer power of toroidal strip-wound cores of VITROVAC 6030F, Fix 350“, VAC Hanau, 1993
- [M3] Simonyi, „Theoretische Elektrotechnik“, 9. Auflage, VEB Deutscher Verlag der Wissenschaften, Berlin, 1989
- [M4] Philippow, „Taschenbuch Elektrotechnik Band 3“, 3. Auflage, VEB Verlag Technik, Berlin, 1989
- [M5] Kampczyk, Röß, „Ferritkerne“, 1. Auflage, Siemens-Verlag, München, 1978
- [M6] Grafe u. a., „Grundlagen der Elektrotechnik, Band 2, Wechselspannungstechnik“, 8. Auflage, VEB Verlag Techni, Berlin, 1980
- [M7] H. Kleinrath, „Grundlagen elektrischer Maschinen“, Akademische Verlagsgesellschaft, Wiesbaden, 1975
- [M8] O. Macek, „Schaltnetzteile, Motorsteuerungen und ihre speziellen Bauteile“, Hüthig-Verlag, Heidelberg, 1982
- [M9] J. Thottuvelil, T. Wilson, H. Owen, „High-Frequency Measurement Techniques for Magnetic Cores“, IEEE Transactions on Power Electronics Vol. 5, January 1990
- [M10] Hirschmann, Hauenstein, „Schaltnetzteile“, Siemens Verlag, München, 1990
- [M11] R. Feldkeller, „Theorie der Spulen und Übertrager“, 5. Auflage, Hirzel Verlag, Stuttgart, 1971
- [M12] Kilgenstein, „Schaltnetzteile in der Praxis“, Vogel-Verlag, Würzburg, 1992
- [M13] S. Cuk, „Survey of Switched-Mode Power Supplies“, IEE International Conference on Power Electronics and Variable-Speed Drives, 1984, pp. 83-94
- [M14] N.N., „Halbleiter“, Siemens AG, München, 1990

- [M15] SGS Thomson, „Handbuch I, Schalttransistoren“, SGS Thomson, München, 1979
- [M16] SGS Thomson, „Handbuch II, Transistoren in der Leistungselektronik“, SGS Thomson, München, 1983
- [M17] J.-P. Vandelaer, P. Zodiak, „A Novel Approach for Minimizing High-Frequency Transformer Copper Losses“, IEEE Transactions on Power Electronics Vol 3, July 1998
- [M18] R. Ridley, C. Zhou, F. Lee, „Application of Nonlinear Design Optimization for Power Converter Components“, IEEE Transactions on Power Electronics Vol 5, January 1990
- [M19] VAC Vacuumschmelze Hanau, „Ringkerne aus VITROVAC<sup>®</sup> 6030F“, Firmenschrift PV-008, Hanau, 8/1989
- [M20] Röß, Hess, Zengler, Rößler, „Ferrite und Zubehör - Fachaufsätze“, Siemens-Verlag, München, 1990
- [M21] L. Kolditz, „Anorganikum Teil 1“, 11. Auflage, VEB Deutscher Verlag der Wissenschaften, Berlin, 1985
- [M22] A. Mulder, „Application note on the design of low profile high frequency transformers“, Philips Components Laboratory Report, 5/1990
- [M23] Philips AG, Valvo, „Ferroxcube 3 und 4 - Allgemeine Angaben“, Philips AG, Valvo, Hamburg, 1987
- [M24] Philips Components, „EFD-Range - Soft Ferrites“, Philips, Eindhoven, 11/1992
- [M25] Philips Components, „3F3 and 3F85, Ferrites for Power Conversion“, Philips, Eindhoven, 1/1990
- [M26] Philips Components, „3F3 Ferrite - at the core of advanced SMPS design“, Technical Publication 282, Philips, Eindhoven, 4/1989
- [M27] Philips Components, „Soft Ferrite Selection Guide“, Philips, Eindhoven, 6/1991
- [M28] R. Boll, „Weichmagnetische Werkstoffe“, 4. Auflage, Siemens-Verlag, München, 1990
- [M29] D. Grätzer, „Weichmagnetische Werkstoffe in modernen Stromversorgungen“, Journal of Magnetism and Magnetic Materials 9, S. 91-98, North-Holland Publishing Company, 1978
- [M30] Harris Semiconductor, „Power MOSFETs, IGBTs“, Harris Corporation, U.S.A., 1992

## **UNIX und das X-Window-System:**

- [U1] P. P. Silvester, „The UNIX System Guidebook, Second Edition“, Springer, New York, 1988
- [U2] S. Strobel, T. Uhl, „Linux - Vom PC zur Workstation: Grundlagen, Installation und praktischer Einsatz“, Springer, Berlin, 1994
- [U3] V. Quercia, T. O'Reilly, „X Window System User's Guide, The Definitive Guides to the X Window System, Volume 3“, O'Reilly & Associates, Sebastopol, California, 1990
- [U4] D. Heller, „Motif Programming Manual, The Definitive Guides to the X Window System, Volume 6“, O'Reilly & Associates, Sebastopol, California, 1991
- [U5] D. Heller, „XView Programming Manual - An OPEN LOOK Toolkit for X11, The Definitive Guides to the X Window System, Volume 7“, O'Reilly & Associates, Sebastopol, California, 1989
- [U6] N. N. „Lesstif Information“, (<http://www.cs.uidaho.edu:8000/hungry/microshaft/lesstif.html>)
- [U7] R. Barthel, „Das VMS-Buch“, Sybex, Düsseldorf, 1990
- [U8] K. Schröder, „UNIX - Das System und seine Handhabung“, 2. neu bearb. Aufl., expert verlag, Ehningen, 1989
- [U9] J. Davidson, „An Introduction to TCP/IP“, Springer, Berlin, 1988
- [U10] U. O. Pabrai, „UNIX Internetworking“, Artech House, London, 1993
- [U11] N. N., „Sun AnswerBook“, Solaris Software Distribution
- [U12] W. M. Newman, R. F. Sproull, „Principles of Interactive Computer Graphics“, second edition, McGraw-Hill, Hamburg, 1983
- [U13] A. Nye, „X Protocol Reference Manual, The Definitive Guides to the X Window System, Volume 0“, O'Reilly & Associates, Sebastopol, California, 1989
- [U14] A. Nye, „Xlib Programming Manual, The Definitive Guides to the X Window System, Volume 1“, O'Reilly & Associates, Sebastopol, California, 1990
- [U15] D. Heller, T. O'Reilly, „X Toolkit Intrinsic Programming Manual, The Definitive Guides to the X Window System, Volume 4“, O'Reilly & Associates, Sebastopol, California, 1990
- [U16] R. Klute, „X11R6 - Die neue Version des X Window System“, iX Multiuser Multitasking Magazin, Heise, Hannover, 7/1994
- [U17] M. Sloman, J. Kramer, „Verteilte Systeme und Rechnernetze“, Hanser, München, Prentice Hall, London, 1989

- [U18] K. Autenrieth, H. Dappa, M. Grevel, W. Kubalski, T. Bartsch, „Technik verteilter Betriebssysteme - Konzepte, Trends, Realisierungen“, Hüthig, Heidelberg, 1990
- [U19] I. Trommer, „UNIX-System V“, Franzis, München, 1988

**Programmiersprachen:**

- [P1] C. J. Sass, „BASIC Programming and Applications“, Allyn and Bacon, Boston, 1976
- [P2] N. N., „American National Standard Programming Language FORTRAN“, ANSI, New York, 1978
- [P3] B. W. Kernighan, D. M. Ritchie, „Programmieren in C“, Hanser, München, 1983
- [P4] S. B. Harbison, G. L. Steele Jr., „C - Ein Referenzhandbuch“, Wolf-ram's Fachverlag, Flensburg, 1991
- [P5] B. Stroustrup, „Die C++-Programmiersprache“, 2., überarb. Aufl., Addison-Wesley, München, 1992
- [P6] B. Stroustrup, „What is object-oriented programming?“, IEEE Software, pages 10-20, May 1988
- [P7] W.-M. Kähler, „SQL - Bearbeitung relationaler Datenbanken“, Vieweg, Braunschweig, 1990
- [P8] N. N., „Allegro CL User Guide - Version 4.1“, Franz Inc., 1992
- [P9] D. Weinreb, D. Moon, „LISP Machine Manual“, Symbolics Inc., Cambridge, Massachusetts, 4. edition, July 1981
- [P10] R. P. Gabriel, „Performance and Evaluation of Lisp Systems“, MIT Press, Cambridge, Massachusetts, 1986
- [P11] K. C. Loudon, „Programmiersprachen - Grundlagen, Konzepte, Entwurf“, Thomson Publishing, 1994
- [P12] P. H. Winston, B. K. Horn, „LISP“, (dt.), Addison-Wesley; Bonn (1988)
- [P13] G. L. Steele Jr., „CommonLISP The Language“, Digital Press, Bedford, Massachusetts, 1984
- [P14] G. L. Steele Jr., „CommonLISP The Language, Second Edition“, Digital Press, Bedford, Massachusetts, 1990
- [P15] H. Bretthauer, J. Kopp, „The Meta Class System MCS - A Portable Object System for Common Lisp - Version 1.3 - Documentation“, GMD, Sankt Augustin, June 1991

- [P16] S. E. Keene, „Object-Oriented Programming in Common Lisp. A Programmer's Guide to CLOS“, Addison-Wesley, Reading, Massachusetts, 1989
- [P17] R. Cordes, R. Kruse, H. Langendörfer, H. Rust „Prolog, eine methodische Einführung“, Vieweg, Wiesbaden (1988)
- [P18] I. Bratko, „PROLOG - Programmierung für künstliche Intelligenz“, Addison-Wesley, Bonn, 1987
- [P19] N. N., „Object Oriented Programming Language Becomes First ANSI Standard“, (<http://www.franz.com/PR/ANSI.html>)
- [P20] B. Haible, „CLISP“, Manualseite
- [P21] N. N., „gcl - GNU CommonLISP“, Manualseite, Free Software Foundation

### **Theorie der Expertensysteme:**

- [X1] N. H. C. Thuy, P. Schnupp, „Wissensverarbeitung und Expertensysteme“, Oldenbourg, München (1989)
- [X2] F. Puppe, „Einführung in Expertensysteme“, Springer, Berlin (1991)
- [X3] P. Harmon, D. King, „Expertensysteme in der Praxis, Perspektiven - Werkzeuge - Erfahrungen“, 3. Aufl., Oldenbourg, München (1989)
- [X4] P. Harmon, D. King, „Expert Systems - Artificial Intelligence in Business“, John Wiley & Sons, 1985
- [X5] P. Harmon, R. Maus, W. Morrissey, „Expertensysteme: Werkzeuge und Anwendungen“, Oldenbourg, München (1989)
- [X6] P. Harmon, D. King, „Expertensysteme“, 2., erg. Aufl., Oldenbourg, München (1987)
- [X7] M. Curth, A. Bölsch, B. Raschke, „Entwicklung von Expertensystemen“, Hanser, München (1991)
- [X8] J. Retti, „Artificial Intelligence“, Teubner, Stuttgart (1986)
- [X9] H.-W. Guesgen, „CONSATS: A system for constraint satisfaction. Research Notes in Artificial Intelligence“, Morgan Kaufman, San Mateo, 1989
- [X10] L. Zadeh, „Fuzzy Logic and Approximate Reasoning“, Synthese 30, pp. 407-428, 1975

### **Expertensystemschemen:**

- [S1] S. Wolf, R. Setzer, „Wissensverarbeitung mit KEE - Einführung in die Erstellung von Expertensystemen“, Oldenbourg, München, 1991

- [S2] T. Christaller, F. di Primio, A. Voss (eds), „The AI-Workbench Babylon. An open and portable development environment for expert systems“, Academic Press, London, 1992
- [S3] T. Christaller, F. di Primio, A. Voss (Hrsg.), „Die KI-Werkbank Babylon: Eine offene und portable Entwicklungsumgebung für Expertensysteme“, Addison-Wesley, Bonn, 1989
- [S4] N. N., „*babylon* Version 3.0 - Softwarewerkzeuge für industrielle Anwendungen“, VW-GEDAS, Berlin, 1991

#### **Literatur zum Expertensystem TRAFEX:**

- [T1] H. Groschwitz and J. Holtz, „An Expertsystem on Transformer Design for SMPS“, IECON, Bologna 1994
- [T2] S. M. Chhaya, B. K. Bose, „Expert System Based Automated Simulation and Design Optimization of a Voltage-Fed Inverter for Induction Motor Drive“, IECON Hawaii, Vol. 2, 1993, pp. 1065-1070
- [T3] L. Gerbaud, J. Bigeon, G. Champenois, „Expert System Bases to Automate Selection of Drive Structures“, IECON Hawaii, Vol. 1, 1993, pp. 360-365
- [T4] D. Chen, B. K. Bose, „Expert System Based Automated Selection of Industrial AC Drive“, IAS Houston, Vol. 1, 1992, pp. 387-392
- [T5] S. Cuk, „Survey of Switched-Mode Power Supplies“, IEE International Conference on Power Electronics and Variable-Speed Drives, 1984, pp. 83-94
- [T6] W. Hirschmann, A. Hauenstein, „Schaltnetzteile“, Siemens AG, Berlin and Munich, 1990
- [T7] N. N., „Halbleiter“, Siemens AG, Berlin and Munich, 1990
- [T8] D. Grätzer, "Soft Magnetic Materials in Modern Power Supplies", Journal of Magnetism and Magnetic Materials, 9/1978, pp. 91-98
- [T9] N. N., „Approximation Formula for the Maximal Transformer Power of Toroidal Strip-Wound Cores of VITROVAC 6030F, Fix 350“, VAC Hanau, 1993
- [T10] S. A. Mulder, „Application Note on the Design of Low Profile High Frequency Transformers“, Philips Components Laboratory Report, Philips, 1990
- [T11] F. Filippetti, M. Martelli, „Development of Expert System Knowledge Base to On-Line Diagnosis of Rotor Electrical Faults of Induction Motors“, IAS Houston, Vol. 1, 1992, pp. 92-99



- [T12] K. Debebe, V. Rajagopalan, „Diagnosis and Monitoring for AC Drives“, IAS Houston, Vol. 1, 1992, pp. 370-377
- [T13] S. Schlums, „Expertensystem für die Bemessung induktiver Bauelemente in primärgetakteten Schaltnetzteilen auf der Grundlage der Shell Babylon“, Diplomarbeit, Universität Wuppertal, 1994

**Sonstige Referenzen:**

- [G1] H.-J. Klein, „Entstehung, Ausbreitung und Wirkung der Störströme von Pulsstromrichtern auf Bahnfahrzeugen mit Wechselspannungseinspeisung“, Dissertation, Universität Wuppertal, 1987
- [G2] J.-O. Kraß, „Maßnahmen zur Verminderung von Oberschwingungsströmen beim Betrieb von Drehstromlokomotiven am Wechselspannungsbahnnetz“, Dissertation, Universität Wuppertal, 1993
- [G3] N. N., „gnuplot“, Manualseite, Free Software Foundation
- [G4] R. G. Herrtwich, G. Hommel, „Nebenläufige Programme“, Zweite Auflage, Springer-Verlag, Berlin, 1994
- [G5] Patzelt, „Windows-Programmierung“, Microsoft Corp.
- [G6] N.N., „Microsoft Developer Studio“, Online-Dokumentation, Microsoft Corp.

# **Eine Diskussion der Problemkreise und Lösungswege bei der Realisierung eines dialoggesteuerten Expertensystems zur Auslegung primärgetakteter Schaltnetzteile**

von Dipl.-Ing. Hans Groschwitz

## **1. Einführung in den Problemkreis**

Schaltnetzteile werden heute in der Stromversorgung vieler elektrischer Anlagen eingesetzt. Im Vergleich zu linear geregelten Netzteilen zeichnen sie sich durch geringere Verluste und somit Betriebskosten, sowie einem geringeren Einsatz an Magnetmaterial aus. Die Kosten eines Netzteils bestimmen sich zu einem großen Teil aus den Material- und Herstellungskosten der benötigten induktiven Bauelemente.

Mit zunehmender Frequenz kann das Bauvolumen eines Transformators zur Übertragung einer bestimmten Leistung reduziert werden. Mit dem Bauvolumen sinken der Einsatz an Magnetmaterial und Kupfer, wie auch der Wickelaufwand. Eine höhere Frequenz als die des speisenden Netzes läßt sich durch den Einsatz von Halbleiterschaltern erreichen. Mittels dieser Halbleiterschalter wird die zunächst gleichgerichtete Netzspannung hochfrequent wechselgerichtet. Das bedeutet, daß durch den Einsatz kostengünstiger Halbleiterbauelemente sowohl die Herstellungs-, als auch die Betriebskosten der Stromversorgung verringert werden können.

Die unterschiedlichen Magnetmaterialien für derartige Mittelfrequenzübertrager werden in einer Vielzahl genormter Bauformen angeboten. Aufgrund produktionstechnischer Abwägungen wie der Lagerhaltung kann diese Grundgesamtheit aber bereits firmenintern eingeschränkt worden sein. Bei der Optimierung hinsichtlich Kosten oder Baugröße hat dies Diskretisierungen und damit Unstetigkeiten der Lösungsräume zur Folge. Thermische Probleme bei der Wärmeabführung, aber auch die notwendigen Wickeldrahtquerschnitte begrenzen eine Reduzierung des Bauvolumens. Das gewählte Schaltnetzteilprinzip beeinflußt Art und Anzahl der Wicklungen. Weitere nichtlineare Einflußgrößen wie die Hysterese des Magnetmaterials schaffen Lösungsräume, die miteinander hinsichtlich des Optimierungsziels konkurrieren. Daher ist auch für erfahrene Entwickler die Auslegung eines Schaltnetzteils zeitaufwendig.

Die Stückzahlen entscheiden im industriellen Alltag oft über den Aufwand, der in die Entwicklung und damit die Optimierung eines Schaltnetzteils gesteckt wird. Es gilt also nicht, das absolute Optimum zu finden, um sich daraufhin einen geeigneten Kern anzufertigen. Vielmehr müssen die verfügbaren Komponenten optimal eingesetzt werden. Die Nichtlinearitäten der Kennlinien von Magnetmaterialien sind zumeist nur näherungsweise bekannt. Insbesondere sind entwicklungsbegleitend thermische Reihenuntersuchungen notwendig.

Die Hersteller der Magnetmaterialien bieten unterschiedliche Lösungswege zur Einsatzbestimmung ihrer Produkte an, die den Kunden, also den Schaltnetzteilentwickler, mehr oder weniger komfortabel an sein Ziel führen sollen. Dabei spielen Erfahrungen der dortigen Entwicklungsabteilungen eine Rolle, sowie die jeweilige Firmenpolitik, die Sicht auf die eigenen Kunden, aber auch das Bemühen, den Mitbewerbern am Markt geringstmöglichen Einblick in Firmengeheimnisse zu eröffnen. Dem Kunden wird ein Lösungsweg vorgegeben, der nur ein Mindestmaß an Transparenz gewährt. Beispielsweise werden Kennlinienverläufe als Tabellen angeboten, so daß Zwischenwerte interpoliert werden müssen. Andere Hersteller stellen geschlossene Näherungsformeln zur Verfügung. Teilweise existieren auch grafisch unterstützte Softwareanwendungen, die einen interaktiven Zugang zu diesen Näherungsformeln oder Tabellendaten bieten. Weiterhin bedeutet die Beschränkung auf die Daten, die für den empfohlenen Lösungsweg benötigt werden, daß die nichtgenannten Parameter Schwankungen unterliegen dürfen. Dies läßt höhere Exemplarstreuungen bei der Serienfertigung oder Modifikationen des Herstellungsprozesses zu.

Allgemein läßt sich feststellen, daß Datenblätter mit Parametern, die sich aus allgemeingültig herleitbaren physikalischen Formeln ergeben und die einen direkten Vergleich zuließen, eher selten zu finden sind [1, 2]. Für den Entwickler, der eine Auswahl aus Produkten der unterschiedlichsten Hersteller vornehmen soll, bedeutet dies, daß er auf der Suche nach dem geeignetsten Kern oft auch den Auslegungsalgorithmus wechseln muß. Aufgrund der vorgeannten Diskretisierungen, die in Unstetigkeiten münden, und der Nichtlinearitäten ist der zeitaufwendige Auslegungsvorgang darüberhinaus über die Gesamtheit der zu untersuchenden Kerne und Schaltnetzteilprinzipien iterativ zu wiederholen.

Im Bemühen um reduzierte Entwicklungszeiten bei möglichst optimaler Auslegung des späteren Schaltnetzteils wird ein Werkzeug gesucht, das herstellerübergreifende Assistenz bei der Transformatorauslegung bieten kann. Um den Interessen der Benutzer gerecht zu werden, sollte dieses Werkzeug eine intuitiv erlern- und bedienbare Software sein. Diese ermöglicht es dem Entwickler, sich auf seine eigentliche Aufgabe zu konzentrieren. Er gibt zunächst die Eckdaten des Schaltnetzteils ein und trifft gegebenenfalls aufgrund seines Erfahrungswissens einige Vorentscheidungen. Die Maschine erledigt die zeitraubenden Berechnungen und präsentiert Ergebnisse, aus denen der Entwickler dann geeignet auswählen kann. Um Einarbeitungszeiten in das Werkzeug selbst gering zu halten, empfiehlt sich die Verwendung zeitgemäßer Mehrfenstertechnik, die es erlaubt, den Benutzer durch die notwendigen Eingaben zu führen, sowie Ausgaben auch in grafisch aufbereiteter Form bereitstellen zu können.

## 2. Ziel der Arbeit

Das Ziel dieser Arbeit ist es, einen geeigneten softwaretechnischen Weg zu erarbeiten, der es ermöglicht, einem Entwickler bei der Auslegung primärgetakteter Schaltnetzteile Assistenz bei der Wahl des geeignetsten Transformators zu bieten. Dazu werden zunächst im ersten Kapitel der Transformator in seinem Umfeld in der Gesamtanordnung Netzteil einschließlich der gegenseitigen Wechselwirkungen betrachtet. Daraufhin wird der Transformator selbst als eine Gesamtanordnung bestehend aus Kern, Spulenkörper und Wicklungen gesehen und diese Komponenten jeweils für sich, sowie als Bestandteil dieser Gesamtanordnung untersucht. Dabei fällt dem Kern mit seiner Fülle an Materialparametern ein besonderes Gewicht zu. Es zeigt sich, daß eine Herleitung allgemeiner Gleichungen nicht praktikabel ist, da die Hersteller von Kernen die dafür notwendigen Parameter nicht in ausreichender Weise bereitstellen. Daher werden Vorgehensweisen zur Transformatorauslegung aufgezeigt, die von den Herstellern von Magnetkernen unterstützt werden und die demgemäß auch zu auswertbaren Ergebnissen führen. Somit steht das für eine erfolgreiche Auslegung notwendige Fachwissen bereit, das der Entwicklungsingenieur bei einer Auslegung ohne maschinelle Unterstützung verwendet hätte.

Bevor dieses Fachwissen zielgerichtet strukturiert werden kann, ist eine Analyse des zu verwendenden Softwarewerkzeuges geboten. Im Kapitel 2 werden zunächst Kandidaten von Werkzeugen vorgestellt und bewertet, um daraufhin die Vor- und Nachteile einer Entscheidung zugunsten einer Neuentwicklung, der Verwendung eines Fertigungssystems sowie teilfertiger Lösungen zu diskutieren. Dabei stellt sich heraus, daß Expertensysteme auf der Basis hybrider Expertensystemschalen der geeignetste Weg sind, die gestellte Aufgabe umzusetzen. Wissensbasierte Systeme sowie Werkzeugumgebungen für deren Erstellung existieren seit Jahren in unterschiedlichen Ausprägungen.

Im Kapitel 3 dieser Arbeit wird ein Ansatz vorgestellt, der es ermöglicht, eine leistungsfähige Wissensverarbeitungseinheit mit geringfügigem Änderungsaufwand um eine grafische Mensch-Maschine-Schnittstelle zu erweitern. Die Geringfügigkeit des Aufwands bezieht sich auf die Änderungen an der wissensverarbeitenden Einheit. Der Ansatz basiert auf einer nebenläufigen Aufgabenteilung zwischen Grafikanwendung und Wissensverarbeitung. Er setzt präemptive Multitaskingbetriebssysteme voraus. Dieser Ansatz ist verallgemeinerbar. Durch ihn läßt sich eine Vielzahl im Einsatz befindlicher komplexer Anwendungen ohne aufwendige Quelltextreorganisation grafikfähig machen.

Anhand einer konkreten Implementierung einer Expertensystemschaale wird die Leistungsfähigkeit eines nebenläufigen Softwarekonzepts unter Beweis gestellt, das eine aufwendige Restrukturierung großer Softwareanwendungen zum Zwecke der Erweiterung um eine grafische Benutzerschnittstelle überflüssig macht. Die Tragfähigkeit dieses Ansatzes wird exemplarisch verdeutlicht, indem eine vorhandene und für den Betrieb an Textterminals ausgelegte hybride Expertensystemschaale um eine leistungsfähige grafische Mensch-Maschine-Schnittstelle erweitert wird.

In Kapitel 4 wird schließlich die Umsetzung der Vorgehensweisen und Rechenvorschriften zur Transformatorauslegung diskutiert. Nach einer Zielfestlegung wird das bereitgestellte Fachwissen zunächst allgemein den unterschiedlichen Wissensrepräsentationsformen hybrider Expertensystemschalen zugeordnet.

Analog dem Vorgehen eines mit der Wissensimplementierung befaßten Softwareingenieurs wird ein Konzept für eine Teilautomatisierung der Aufgabe mit dem Ergebnis einer starken Entlastung des Schaltnetzteilentwicklers aufgezeigt. Dialoggesteuert wird zunächst mit Unterstützung aus der Anwendung das Anforderungsprofil an das zu realisierende Netzteil eingepflegt. Daraufhin wird die Netzteiltopologie bestimmt. Auf der Basis einer Kerndatenbank werden Kandidaten ermittelt, sowie Wickelvorschriften ausgegeben. Diese Ergebnisse lassen sich gegeneinander bewerten. Nötigenfalls lassen sich durch eine Anpassung der Startbedingungen befriedigendere Ergebnisse erzielen.

Basierend auf diesem Ansatz ist exemplarisch ein Expertensystem entwickelt worden, um bei der Auslegung des Mittelfrequenztransformators für eine gegebene Schaltnetzteiltopologie zu unterstützen. Darin wird für das zu entwickelnde Schaltnetzteil aus einer Datenbank ein optimaler Kern ausgewählt. Dabei werden Kriterien wie die Leistungsbemessung, Ausgangsspannung, thermische Bemessung, das Bauvolumen und die Isolationsklasse berücksichtigt. Zur Ermöglichung benutzerfreundlich gestalteter interaktiver Sitzungen ist eine grafische Benutzerschnittstelle erstellt worden.

## 3. Schrifttum

- [1] Siemens AG, „Ferrite und Zubehör Datenbuch 1990/91“, Siemens AG, 1989
- [2] Vakuumschmelze Hanau, „Approximation formula for the maximal transformer power of toroidal strip-wound cores of VITROVAC 6030F, Fix 350“, VAC Hanau, 1993